ROBOTICS MICROCONTROLLERS COMPUTER CONTROL SPACE

Everything For Electronics

Nuts & Volts

October 2004
www.nutsvolts.com

RADIO PARADE!

Your Transistorized Pal Turns 50

LOOK INSIDE FOR:

- Digital PID Control
- Secrets of Serial Debugging
- Non-Contact Distance Measuring

TABLE OF CONTENTS

Vol. 25 No. 10

FEATURES

Cover by Jean Von Trende Photography
www.jvtphotography.com

PROJECTS

COLUMNS

Note to our readers: In order to bring you more of the projects you have requested, we have made the "Personal Robotics" and "Near Space" columns bimonthly. Look for "Near Space" in the November issue and "Personal Robotics" following in December.

OCTOBER 2004

p. 71

p. 66

p. 94

p. 24

DEPARTMENTS

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree#40702530) is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, 430 Princeland Court, Corona, CA 92879-1300** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com


SUPERCIRCUITS
WWW.SUPERCIRCUITS.COM
VIDEO CRIME STOPPERS AND JAW DROPPERS!!!
110% LOW PRICE GUARANTEE
*Some restrictions apply. See catalog or website for details.
QUALITY SERVICE INNOVATION 15 Years
OUTDOOR FLOOD LIGHT CAMERA
Powered by and transmits video through AC lines!
Receiver Included
$99.95
TINY MICRO VIDEO TRANSMITTER
Capable of up to 1000 foot transmission distances and is smaller than your thumbnail.
$189.95
STEALTHY CLOCK RADIO WITH HIDDEN WIRELESS COLOR VIDEO CAMERA
SUPER EASY TO USE!
300 foot range, includes receiver!
$109.95
COLOR INFRARED WEATHERPROOF CAMERA
High Intensity LEDs Let You See Up To 45 Feet In Total Darkness!
$109.95
RV & TRUCK VIDEO REAR VIEW CAMERA SYSTEM
Includes camera, monitor, cables & mounts!
$99.95
WORLD'S SMALLEST CAMERA
Featured in the Guinness Book of World Records!
Tiny 0.375" square by 0.625" size!
$99.95
B/W MICROVIDEO CAMERA
UNDER $12
1.25"
Only $11.95—World's Best Value Video Camera!
Put it on your R/C plane with our video xmitter for a bird's eye view!
$11.95
MINI VIDEO SERVER
Only $249.95!
View any camera or video signal remotely anywhere in the world with standard internet browsers
$249.95
2.4 GHZ WIRELESS LINK—4 CHANNELS & 700 FOOT RANGE
Includes transmitter, receiver, & power supplies
$109.95
UNDERWATER VIDEO SYSTEM WITH MONITOR AND BATTERIES
65 foot cable!
$149.95
1280 HOUR TIME LAPSE VCR
Only $179.95!
This Time Lapse VCR Lets You Record 1280 Hours--That's Nearly 2 Months!
$179.95
QUAD/DVR VIDEO CARD
View and record video up to 4 cameras on your PC!
Records up to 1000 hours of video on an 80 GB hard drive!
$89.95
1" MINI MONITOR
New! Handheld pocket size mini field monitor!
Runs up to 24 hours on 3 AAA batteries!
$119.95
NEW! MICRO DIGITAL VIDEO RECORDER
Records up to 40 Hours!
World's smallest hard-drive based digital video recorder!
WORLD'S SMALLEST
$499.95
WIRED COVERT SMOKE DETECTOR CAMERAS
Covers almost any room with 30º oblique camera angle
Covert camera location
Ask about or super stealth versions for phenomenal low light rating of 0.0003 lux.
700 foot wireless version available complete with receiver and power supplies for $189.95
$59.95
PRO 4 CAMERA QUAD VIDEO SECURITY SYSTEM
Complete system, great for stores!
It's easy to set up a complete, full-featured professional video monitoring security system with 4 of our PC-152C video cameras and lenses, a realtime quad processor, a high resolution 12" monitor, and much more.
$399.95
MICRO AUDIO SYSTEM
PLUG & PLAY
Super high gain microphone with built in preamplifier!
$9.95
WEATHERPROOF OUTDOOR CAMERA WITH ZOOM LENS
Installation is a snap with easy external magnetic zoom adjustment.
PERFORMANCE • VALUE • FEATURES • WARRANTY Signature SUPERCIRCUITS Series
Comes with versatile 5-50 MM varifocal zoom/auto iris lens.
$219.95
SUPERCIRCUITS
See Hundreds Of Amazing Video Products Online At www.supercircuits.com or Call Us Today To Order or Request A Free Catalog
1-800-335-9777
One Supercircuits Plaza, Liberty Hill, TX 78642
Phone 1-512-778-6950 Fax 1-866-267-9777
Prices, specifications and availability are subject to change without notice. Published terms & conditions apply. Copyright 2004 Supercircuits, Inc.
Please call us at 1-800-335-9777 to be rushed your free Microvideo and Security Catalog!

Reader Feedback

Dear Nuts & Volts:

I am writing about "The Stereo 6T9" tube amplifier in the August issue of *Nuts & Volts*. I have a couple of suggestions.

Regarding negative feedback, there is a simple way to use both the bypass caps and negative feedback. Connect a 100 ohm resistor between R102 and ground. Connect a bypass cap in parallel with R102 (the cap is not connected to ground). Because R102 is larger than R105, a smaller cap can be used here, maybe 33 µF.

Reduce R108 by the ratio of the 100 ohm resistor to R104 (1/22), making R108 1,500 ohms. The value of C105 should be increased similarly. (If too large a value is used for C105, it may produce — rather than prevent — ultrasonic oscillation, especially if a different output transformer is used. If this is a problem, try changing the value of C105. Also try increasing the value of R108.)

Just in case anyone uses the amplifier with a vacuum tube signal source, the volume controls in Figure 6 should have a higher resistance, 250K or 500K ohms.

In addition to Antique Electronic Supply, dual audio taper pots are available in several resistance values from Mouser Electronics 1-800-346-6873, **www.mouser.com**).

Bill Stiles
via Internet

Dear Nuts & Volts:

Thanks for a very good article on Smith Charts! On page 77, the equation for a series inductor and answer are incorrect, though. It should read $X_L = 2\pi \cdot 2.1E6 \cdot L = 0.8 \cdot 50$, which yields an answer of 0.3 µH series.

I also liked the LORAN article and the history of Hugo Gernsback. As an engineer for over 50 years, I am still learning from your magazine.

Robert H. Miller
Garner, NC

Dear Nuts & Volts:

I just read the Reader Feedback in the September 2004 issue, where Bob E. Baker is talking about Ohm's Law. I have to say that I agree with him wholeheartedly, but — instead of waiting for someone else to write the article — why doesn't Bob do it? C'mon, Bob, do it!

T. Morris
via Internet

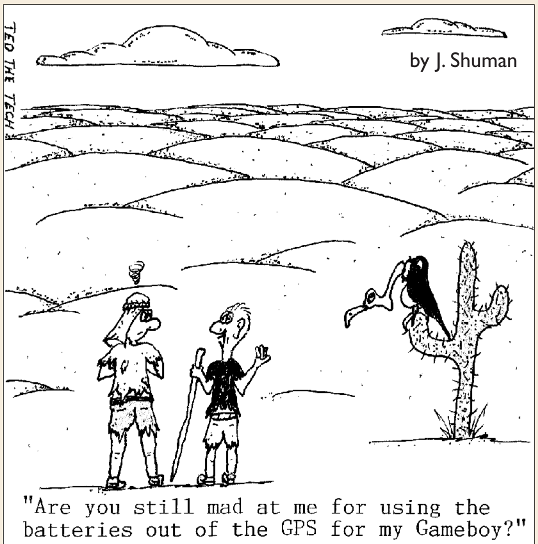
Dear Nuts & Volts:

The article on the Stereo 6T9 tube amplifier was just spot on! My personal hobbyist bias is slanted toward analog electronics, anyway because — as a software developer — I work with the digital world every day.

For those of us who have grown up in the transistor age, there's a bit of mystery and a certain retro nostalgia for the old vacuum equipment. Plus, anyone with a bit of audiophile familiarity or ham radio knowledge is aware of the high power vacuum tube-based equipment that is in vogue in those circles, at least in the high end sphere.

Thanks for publishing a project article that puts a vacuum tube experience within reach of the rest of us.

Roger Voss
Maple Valley, WA



"Are you still mad at me for using the batteries out of the GPS for my Gameboy?"

Dear Nuts & Volts:

Steve Lawson's comments about the development of white light in his article dealing with white LEDs in the August issue need further clarification. He stated that, "yellow is a combination of red and green."

It is true that the perception of yellow can result

(Continued on Page 41)

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
www.nutsvolts.com
Subscription Order ONLY Line
1-800-783-4624

FOUNDER/ASSOCIATE PUBLISHER
Jack Lemieux

PUBLISHER
Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
robin@nutsvolts.com

MANAGING/TECHNICAL EDITOR
Dan Danknick
dan@nutsvolts.com

ASSOCIATE EDITOR
Alexandra Lindstrom
alexa@nutsvolts.com

CONTRIBUTING EDITORS
Sarah Lowrey, Gerard Fonte, David Caldwell, TJ Byers, Jeff Eckert, Gary Peek, Jon Williams, Walter Krawec, Tom Dickens, Ron Newton, Louis Frenzel, Peter Best, Mark Balch

CIRCULATION DIRECTOR
Mary Descaro
subscribe@nutsvolts.com

SHOW COORDINATOR
Audrey Lemieux

WEB CONTENT/NV STORE
Michael Kaudze
michael@nutsvolts.com

PRODUCTION/GRAPHICS
Shannon Lemieux

STAFF
Janessa Emond
Kristan Rutz

MSP430

It's Time for a Change!

2004 MSP430 Advanced Technical Conference (ATC)

- Experience cutting-edge ultra-low-power technology
- Meet with TI MSP430 experts, third parties and other MSP430 users from around the world
- Witness demonstrations of the highest-performance MSP430 yet
- Sharpen your knowledge of MSP430 in an interactive, informal setting
- Attend in-depth workshops, seminars and presentations on MSP430:
  - Architecture
  - Peripherals
  - Coding/design techniques
  - Applications

Seating is limited, register today at www.ti.com/atc2004

World's Lowest Power MCU

If you're not already using the MSP430 in your design, it's time for a change.

- 0.1-µA RAM retention
- 0.8-µA RTC mode
- 250-µA/MIPS active
- 6-µs clock startup
- 50-nA low-leakage pins
- Vectored interrupts
- DMA
- High-performance analog ideal for precise measurements
- 16-bit RISC CPU enables new applications with less code
- ISP Flash permits flexible code changes
- Complete integrated development environment $99
- Device pricing as low as $0.49

Online Registration
Data Sheets and Samples

www.ti.com/atc2004 ○ 800.477.8924, ext. 1012

REAL WORLD SIGNAL PROCESSING™ TEXAS INSTRUMENTS

by Mark Balch

Basics For Beginners

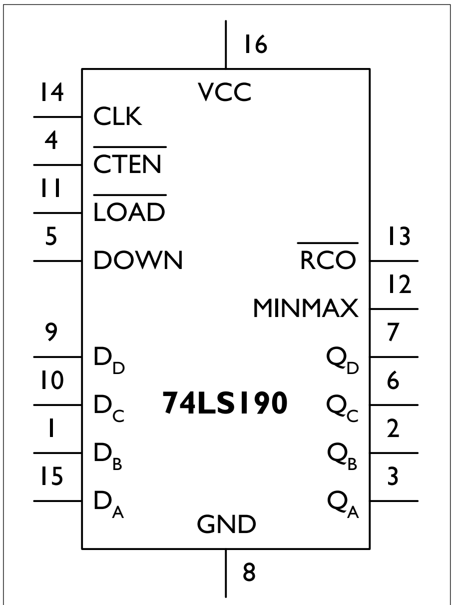
Just For Starters

Building an Event Counter

A reader recently asked me how to design a counter. Counters are a basic digital logic building block and have a multitude of uses. Counters advance through a consecutive numerical sequence — either up or down — each time a clock pulse is driven. They are used to divide high frequency clocks to yield lower frequency clocks, for state machines and basic event counting. You can design a counter from scratch with truth tables by applying Boolean logic. In this article, we will discuss using off-the-shelf counter chips to take advantage of pre-fabricated building blocks.

The common 7400 logic family has numerous four-bit counter chips available. Four bits is a popular size because the counter is small enough to fit in a 16-pin package and can be easily cascaded to form longer widths: two four-bit counters give you a byte counter. A four-bit binary counter normally advances from 0000_2 to 1111_2 or 0 to 15 decimal. This natural power-of-two count sequence is useful in many applications. Some applications, however, are more suited to a decade counter that advances from 0000_2 to 1001_2 or 0 to 9 decimal. Decade counters are often used when counting events that are to be interpreted by people, since people tend to think in decimal. The term binary coded decimal — or BCD — refers to a binary counter that wraps from 1001_2 to 0000_2 rather than continuing with a normal binary count sequence.

Figure 1. 74LS190 pin assignment.



74LS190 Decade Counter

A well-known decade counter is the 74LS190. It has been around for a long time, as evidenced by the original date of December 1972 printed on the data sheet that I downloaded via the web. The 'LS190 is a synchronous counter, meaning that it outputs transition only on a low-to-high clock edge. This gives the counter very predictable behavior.

Figure 1 shows the 'LS190 pin assignment. Aside from two power pins, the chip has four count inputs, four count outputs, a clock, and several control signals. The count inputs, D_N, enable pre-loading the counter with a specific value, giving you the option of not always starting the count from zero. The $\overline{\text{LOAD}}$ signal transfers the D_N state inside the chip, which is reflected at the outputs, Q_N, on the next rising clock edge. We will use this feature to reset the chip to start counting at zero.

Next, there is a DOWN signal that tells the chip to count down when DOWN = 1 or to count up when DOWN = 0.

Aside from the clock, CLK, there is a count enable signal, $\overline{\text{CTEN}}$, that qualifies CLK. When $\overline{\text{CTEN}}$ = 0, the chip increments or decrements normally on the clock's rising edges. When $\overline{\text{CTEN}}$ = 1, the counter does not advance and retains its present value, unless a new value is loaded via the $\overline{\text{LOAD}}$ signal. $\overline{\text{CTEN}}$ and the two outputs, RCO (ripple carry out) and MAXMIN, facilitate cascading multiple counters to yield practically any size counter.

A Cascaded Counter

So how does this all come together? Figure 2 shows a byte-wide cascaded counter comprised of two 74LS190 chips. Note that both chips have common CLK and $\overline{\text{LOAD}}$ signals. This means that both chips can advance together on the same rising clock edge. It also means that they can be reloaded together. The example has D_N tied low so that the counter is reset to 0 when $\overline{\text{LOAD}}$ = 0 and CLK is driven low-to-high. The DOWN signal is set low on both chips to indicate an advancing count sequence from 0 to 9 for each chip.

Finally, the counters are cascaded to form one single counter by driving one counter's $\overline{\text{CTEN}}$ signal with the other's $\overline{\text{RCO}}$ signal. The least significant digit's $\overline{\text{RCO}}$ pulses low each time the count value is 9. This enables the most significant

digit's counter for one clock cycle so that it advances on the next CLK edge.

After this next edge, the least significant digit wraps around to become 0, which restores RCO to its high state and deactivates the most significant digit counter until it is again time to advance.

Control Inputs

At this point, we have the core of a counter circuit, but there are two dangling inputs: CLK and LOAD. These can be manual push buttons so that you can control when the counter is reloaded with 0 and when it advances. LOAD can be directly tied to a push button. CLK, however, is not as simple because it regulates the entire circuit's behavior. CLK must be a clean signal with uniform high-low transitions so that the counter behaves correctly.

If we directly connect a push button to CLK, the counter is likely to behave erratically. The reason for this is that a mechanical button generates electrical noise as its internal surfaces make and break contact. This noise may be imperceptible to a person, but a synchronous logic circuit will react wildly. A debounce circuit (such as the one shown in Figure 3) can fix this problem.

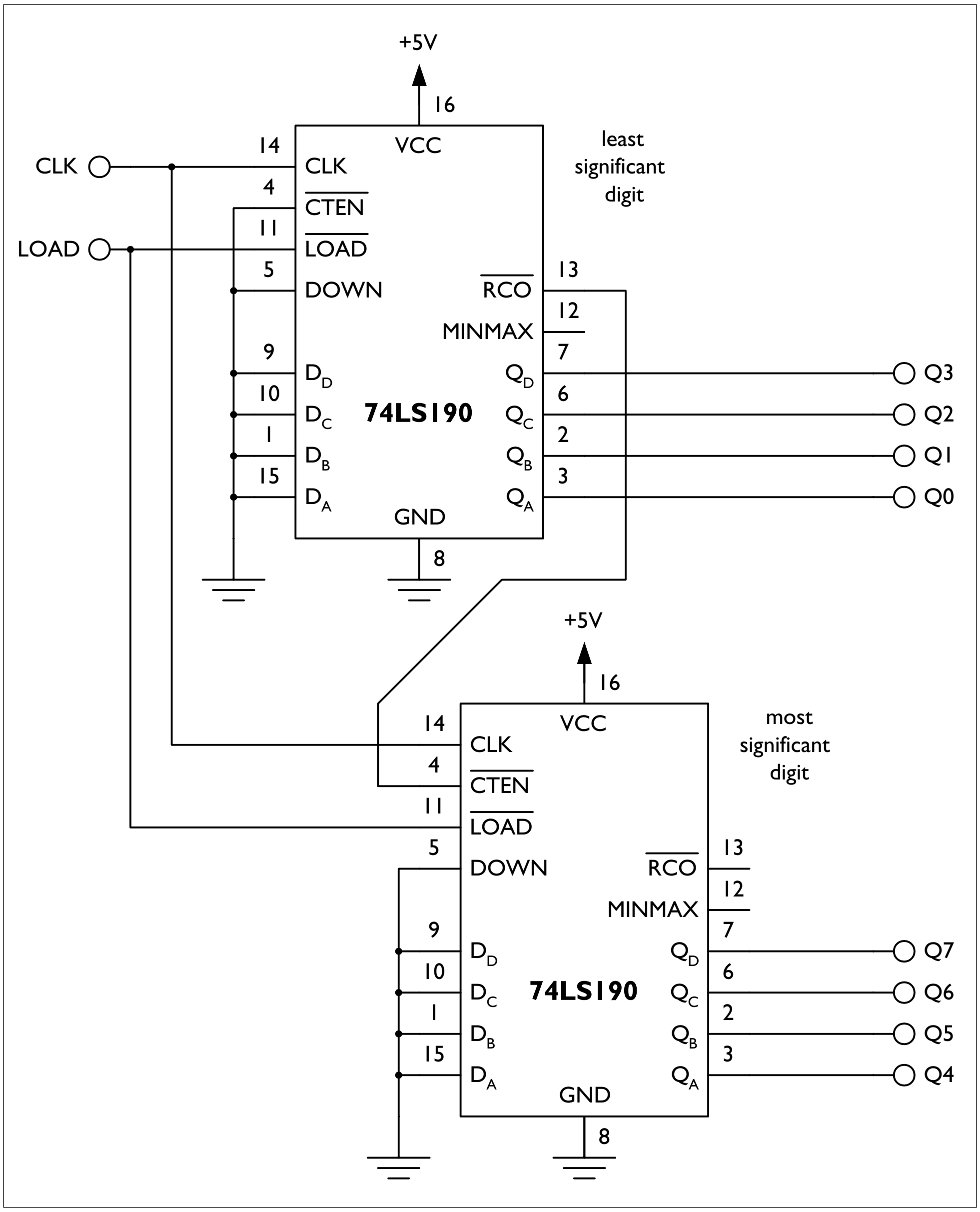


Figure 2. Byte-wide counter.

As its name implies, a de-bounce circuit removes the bounce — or noise — from an input and generates a clean output. There are numerous techniques to de-bounce a push button. The one shown here uses an RC filter along with a Schmitt trigger inverter (made from a NAND gate) to filter out the noisy push button clock signal. The RC time constant is 100 milliseconds when discharging from 5 V to 0 V and 200 milliseconds when charging back to 5 V.

You can adjust the time constant higher or lower to suit your specific needs.

After connecting this portion of the circuit, the counter cleanly increments each time you press the CLK button and reloads when you hold down LOAD and press the CLK button.

Seeing Is Believing

The counter is now functional, but does not give you very much indication that it may be working as it currently stands.

We need to connect a visual indicator to see what count values are being generated. The simplest thing to do is connect an LED to each of the eight outputs and watch the BCD pattern change with each CLK press.

It is more interesting, however, to read a recognizable 0-99 count sequence. For this, we turn to the 74LS47 BCD-to-seven-segment decoder/driver. You may observe seven-segment displays all around you: microwave ovens, digital watches, stereos, and VCRs. Each

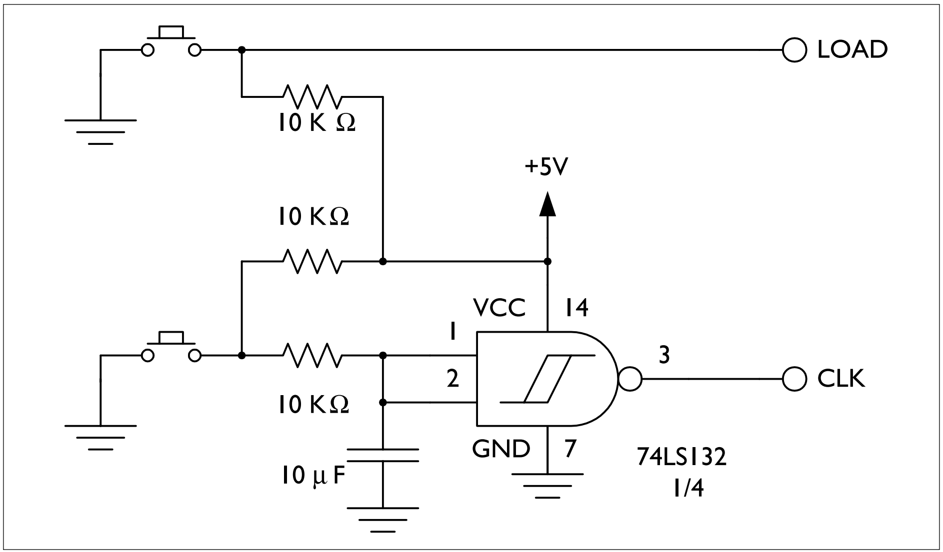


Figure 3. Clock debounce circuit.

display contains seven independent light elements — often LEDs — that can form a recognizable decimal digit. The 74LS47 converts the BCD output of the 74LS190 into a human-readable, seven-segment format.

Figure 4 shows how one 74LS47 chip connects to the least significant digit of the counter that we have already created.

(A second, identical 74LS47 and seven-segment display connect to the most significant digit.)

It is a simple hookup with just three control signals to tie high. The 74LS47 has active-low outputs that are designed for common-anode LED displays.

A common anode display has all of the LED anodes connected to a single pin. The individual cathodes connect to the decoder chip through current limiting resistors. The resistors prevent too much current from flowing through the LED into the 74LS47 and damaging either. Each segment of the seven-segment display is identified as "a" through "g."

Counters and More Counters

There is virtually no limit to what

About the Author

Mark Balch is the author of *Complete Digital Design* and works in the Silicon Valley high tech industry. He can be reached though his website at **www.completedigitaldesign.com**

you can do with counters. They are present — in one form or another — in almost every complex digital logic circuit. You can design custom counters with your own logic or you can employ off-the-shelf counters. You can read more about counters, counter design, current limiting, and LEDs.

Fortunately, there are many resources and components available that will allow you to experiment and build whatever counters you are interested in. **NV**

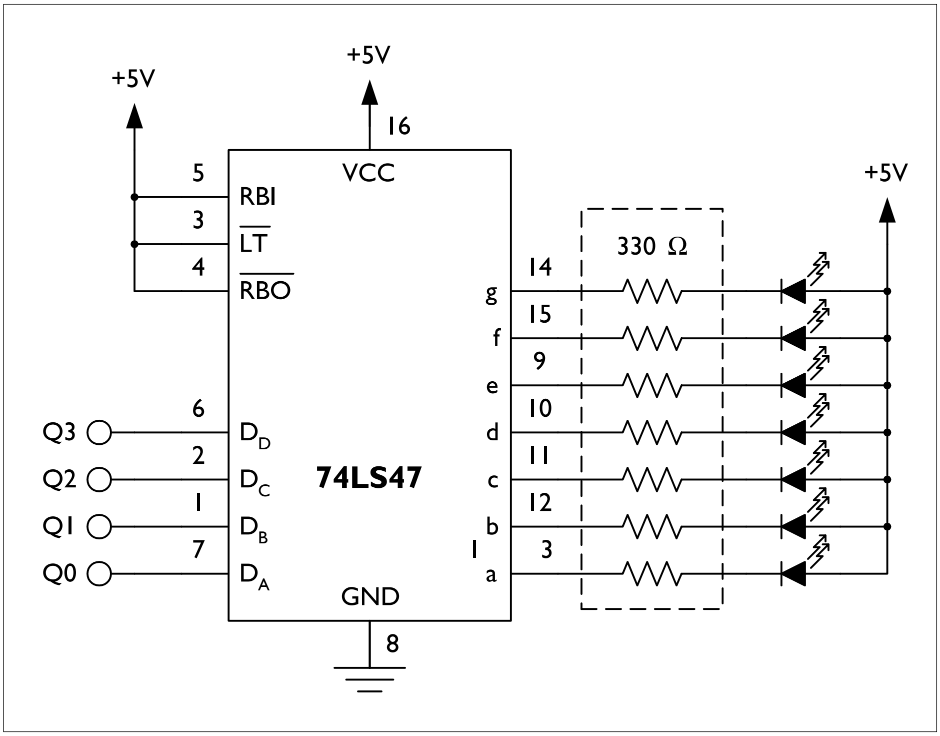


Figure 4. Seven-segment display circuit.

More build-it-yourself High Performance Electronic Kits for Cars

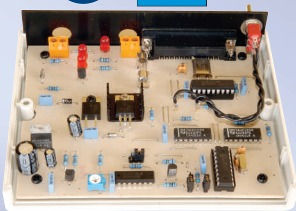
These projects offer fantastic tuning and performance modifications for your car. All projects are from the book High Performance Electronics Projects for Cars - published by Silicon Chip Magazine, Australia's leading Electronics magazine, available soon! See 2004 catalog pages 12 & 13 for details.

Digital Pulse Adjuster Kit

3 hours — COMPLEX

KC-5384 $46.10

A huge revolution in DIY performance! Control and tune the operation of a solenoid that is run by the engine management system. You could alter auto transmission shifts, control an extra injector, and more! Kit supplied with PCB, machined case, and all electronic components. Requires KC-5386 Hand Controller below.



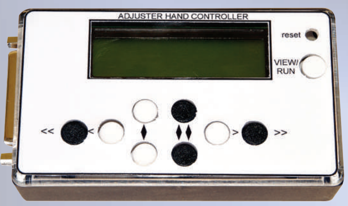
Hand Controller Kit for Digital Adjusters

KC-5386 $34.55

Real time or programming display! This controller is used for all of the digital adjuster kits available. It can be connected for programming then removed, or left connected for real time display. Kit supplied with silk-screened and machined case, PCB, LCD, and all electronic components.

90 mins — COMPLEX

Moreinfo?
www.jaycarelectronics.com

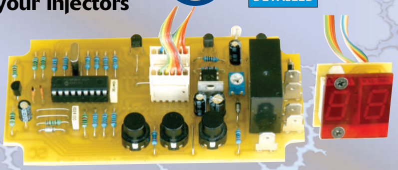


Duty Cycle Meter Kit

60 mins — DETAILED

KC-5375 $31.70

Super fast real time sampling! Automatically cut-in an extra fuel pump when your injectors reach a certain level and more! Includes a simple duty cycle generator for testing. Kit supplied with PCB, and all electronic components.

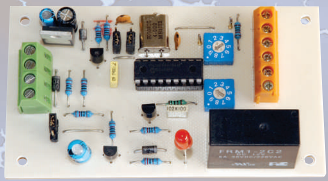


Intelligent Turbo Timer Kit

60 mins — DETAILED

KC-5383 $25.90

There are turbo timers, and there are intelligent turbo timers. It determines how hard the car has been driven, and idles for an appropriate time after ignition switch cutoff. Kit supplied with PCB, and all electronic components.

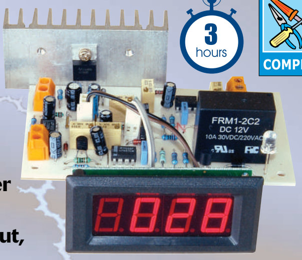


High Range Adjustable Temp Switch Kit with LCD Readout

3 hours — COMPLEX

KC-5376 $40.30

Range up to 2100°F! Keep an eye on critical temperatures such as brakes, turbo manifolds, intercoolers, and more. Trips a relay at a preset level to trigger an alarm, water spray cooling, and more! Kit supplied with PCB, LCD readout, and all electronic components.

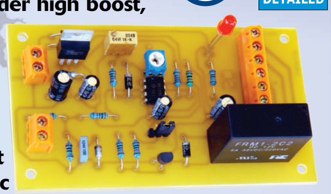


Simple Voltage Switch Kit

60 mins — DETAILED

KC-5377 $17.30

Suits a huge range of applications! Use it to trigger an extra fuel pump under high boost, anti-lag wastegate shutoff, and much more. Can detect rising or falling voltage, and features adjustable hysteresis (the difference between trigger on and off voltage). Kit includes PCB, and all electronic components.

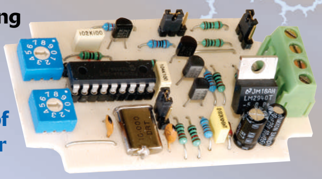


A Cheap Nitrous Fuel Mixture Controller

90 mins — COMPLEX

KC-5382 $14.50

Nitrous oxide systems can be expensive to set up, but now you can do it for much less. This project pulses a fuel injector at a preset rate, adding a fixed amount of nitrous fuel when you activate it. It will save you a bundle on dedicated fuel solenoids and jets. It can also be used to control electronic water pumps, cooling fans, and more. Kit supplied with PCB and all electronic components. *Please check local laws regarding the use of Nitrous Oxide systems in your vehicle.



Explanation of icons used for kits.

DETAILED — The kit requires a good knowledge of electronics to understand its operation, and troubleshooting techniques may be required.

COMPLEX — The kit may require several sittings to complete, and possibly require some mechanical assembly. Troubleshooting techniques a must.

60 mins — Indicates an approximate construction time for each project for a competent constructor. It does not however, include any installation etc that may be required.

ORDER on-line: www.jaycarelectronics.com

Check our website! Information & easy, safe, fast ordering.

- Entire Jaycar 2004 Catalog on-line - over 6,000 products.
- 128-bit Secure on-line ordering - safe & secure.
- Search by category, keyword or catalog number, & advanced search.
- Over 3,500 product datasheets & application notes available on-line.
- And we're from Australia so you can trust us!

MasterCard VISA

www.jaycarelectronics.com

TOLL FREE ORDERING 1-800-784-0263

(We are open 6pm to 2:30am East Coast time, which equates to 9am to 5:30pm our time)

For those that want to write: 100 Silverwater Rd Silverwater NSW 2128 Sydney Australia

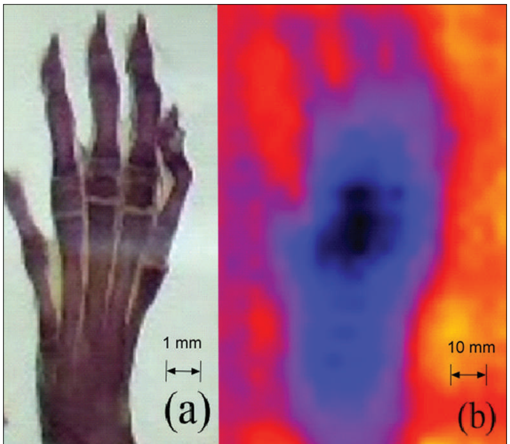
Jaycar Electronics

Circle #81 on the Reader Service Card.

TechKnowledgey 2004

Events, Advances, and News From the Electronics World

Advanced Technologies
New Way to Look at Things



Neutron micrography, such the image of a rat's foot shown at right (b), may offer advantages over existing X-ray and electron imaging techniques (a). Courtesy of Adelphi Technology, Inc.

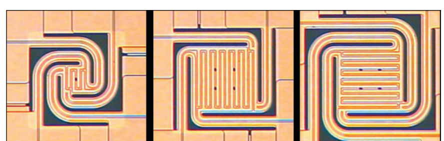
A prototype microscope that uses neutrons instead of light to "see" magnified images has been demonstrated at the National Institute of Standards and Technology (NIST, **www.nist.gov**). The hope is that neutron microscopes may eventually offer advantages over optical, X-ray, and electron imaging techniques, including better contrast for biological samples. Adelphi Technology, Inc. (**www.adelphitech.com**), designed and demonstrated the microscope with the help of NIST scientists, who routinely use multiple lenses to focus neutron beams for other research. Stanford University also participated in the research, which was supported in part by the US Department of Energy.

The imaging process involves hitting a sample with an intense neutron beam. The neutrons that pass through — thereby creating a pattern that reflects the sample's internal structure — are directed to a row of 100 dimpled aluminum plates. Each dimpled plate acts like a weak focusing lens for neutrons, diverting their paths slightly at each interface. The image is then projected onto a detector.

In principle, neutrons could provide better image resolution than visible light because they have shorter wavelengths — as short as 1 nm. In this demonstration, the microscope produced a resolution of only 0.5 mm and a magnification of about 10. However, Adelphi hopes to substantially improve image resolution through research to reduce lens aberrations. The company also hopes to build a compact, laboratory-scale neutron source.

Moreover, neutrons offer some unique advantages. Unlike other imaging methods, neutrons interact strongly with hydrogen — an important component of biological samples composed mostly of hydrocarbons and water. Also, neutrons easily penetrate samples, thereby reducing the artifacts produced with other techniques that require thin slices, staining, or fixing.

Miniature Hotplates Reach 1,100° C



Silicon carbide micro-hotplates, each consisting of a central plate surrounded by curved tethers. The largest is less than 100 µm across.

Engineers at Boston MicroSystems, Inc. (**www.bostonmicrosystems.com**), have come up with a series of micro-hotplates that are only a few dozen micrometers across and can achieve temperatures exceeding 1,100° C (2,012° F). Built with silicon carbide to tolerate the extreme heat, they reach peak temperature in less than 0.001 second. Silicon carbide is not only stable at high temperatures, it is also impervious to chemical attack from most materials. As a result, the hotplates can be cleaned by merely burning debris off the surface.

Contained on a microchip, the hotplates currently are used in tiny "labs" inside a transparent polycarbonate chamber that can endure near-vacuum pressures. Ports on the chamber's sides allow gases to pass through and feed experiments; researchers can observe experiments with a microscope. The hotplates also contain an integrated temperature gauge and a pair of electrodes. These components allow researchers to test the electrical properties of various materials that may be deposited onto the hotplates.

Using the stable, thin-film deposition properties and integrated circuitry of the hotplates, researchers are already developing applications, such as oxygen and engine emission sensors. The sensor may have several advantages over devices in today's combustion engines due to the micro-hotplate's chemical stability, small size, rapid response, and low power consumption. Prospective applications are in such diverse areas as heat treatment analysis, thin film material characterization, automobile emissions and other environmental testing and instrumentation, and biological adaptations.

The techniques for crafting and optimizing these microelectromechanical systems (MEMS) were developed with support from the National Science Foundation Small Business Innovation Research (SBIR) program and SBIR programs at the Department of Energy, Environmental Protection Agency, and NASA.

Computers and Networking

Grand Haven, MI: "Hot City"

It's just a little waterfront town known for sport fishing, its "One Sky, One World International Kite Fly for Peace," and its stewardship of the world's largest musical fountain. Now, the 12,000 residents have one more distinction: Grand Haven is the first city in the US to provide wireless fidelity (WiFi) networking everywhere within the city limits and it lays claim to being the nation's first "hot city."

Developed and managed by Ottawa Wireless (**www.ottawawireless.net**), the system uses several hundred strategically located WiFi (802.11a, b, g) radios to blanket its six square miles and provide coverage 15 miles into Lake Michigan. WiFi gives users the freedom to receive and transmit data over the Internet at high speeds from anywhere within the broadcast signal range. Voice over IP (VoIP) Internet-based phone service is also available on the new network.

In addition to fixed/mobile high speed Internet access, Ottawa Wireless provides point-to-point dedicated VPN connections, remote wireless video surveillance, and high speed access for boaters up to 15 miles offshore and at speeds of up to 55 miles per hour. City-wide mobile WiFi VoIP telephone calling is currently in beta testing, with a full launch expected in the next few months. Monthly prices for always-on broadband Internet starts at $19.99 for 256 kbps and unlimited mobile VoIP calling is $29.99.

Storage Format Announced

If you have been working with computers for a decade or more, you no doubt have acquired quite a few storage devices, some of which now function perfectly as paperweights. Somewhere between your desktop and closet, you may still possess Syquest drives, magneto-optical drives in various sizes and formats, Zips, tape backup devices, and others, plus the more modern CD and DVD technologies.

Well, get ready for the Blue-ray Disc, devised by a consortium that includes Dell, Hewlett-Packard, Hitachi, Sony, TDK, Matsushita, Pioneer, Royal Philips, Samsung, and others, known as the Blue-ray Founders. Version 1.0 of the BD-ROM has been approved and is now available to disk manufacturers.

Apparently designed to compete with the HD-DVD format backed by NEC and Toshiba, it may be coming to a computer near you sometime in 2005.

Driven by blue laser diode technology, the Blue-ray Disc provides storage of 25 GB on a single-layer disk or 50 GB on a dual-layer one. The 12 cm diameter is the same size as a CD/DVD and it offers transfer rates of 36 Mbps. Key applications include recording and playback of high definition video (for which one disk can provide up to four hours of HDTV), but it is also applicable to PC data storage. For details, you can visit the Blue-ray Founders' website at **www.blu-raydisc-official.org**

Circuits and Devices

Single-Chip Car Radio Announced

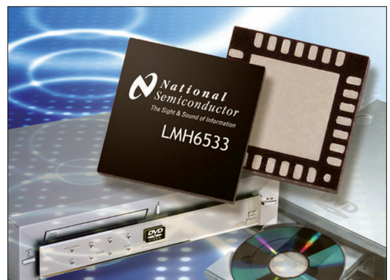
Royal Philips Electronics (**www.semiconductors.com**) recently introduced a family of one-chip analog car radio solutions that promise superior tuning performance and reduced overall system costs in factory-installed automobile radios.

By combining Philips' front-end tuning and analog signal processing technologies, the TEF690x chips are said to reduce the number of external components needed to build high performance car radios and simplify the design process to reduce system costs by 20 to 30 percent as compared to existing two-chip systems.

The TEF690x devices include an AM/FM tuner, stereo decoder, adaptive IF bandwidth control, precision adjacent-channel suppression (PACS), and advanced weak signal processing. In addition, the devices offer flexible input selection and the option of an integrated RDS demodulator and/or the connection of an external sound processor or navigation/beep input. This variable feature set enables car radio manufacturers to serve worldwide markets in Asia, Europe, and the US with a single platform.

The devices are sampling now, with mass production to begin in 2005. Four different devices will be available, including the TEF6902 — which incorporates all the standard features of the TEF690x range into a 64-pin package — and the TEF6901 — which will also feature an integrated RDS demodulator in a 64-pin package. Philips' TEF6903 (with an integrated RDS demodulator) and the TEF6904 will include external processor I/0 and will be available in 80-pin packaging. Pricing was not disclosed as of press time.

Laser Diode Driver Improves Optical Drives



National's LMH6533 is geared for higher performance DVD and CD drives. Courtesy of National Semiconductor.

National Semiconductor Corp. (**www.national.com**) has announced a new laser diode driver (LDD) for use in optical pickup units (OPUs). The LMH6533 offers fast switching rates, very low output current noise, and low power consumption. The reduced power consumption provides better heat dissipation in the system, while the low noise improves read times for optical disk drives. The LMH6533 is designed for combination DVD/CD recorder optical storage devices used in desktop, notebook, and consumer DVD video recorders.

The LMH6533 laser diode driver contains two high current outputs for reading and writing DVD (650 nm) or CD (780 nm) lasers. It achieves read, write, and erase functions through four separate switched current channels. The LVDS interface delivers DVD write speeds of 16x and higher while minimizing noise and crosstalk. The device provides one 300 mA and two 150 mA write channels, plus a 150 mA read channel.

Additionally, it minimizes electromagnetic interference (EMI), allowing disk drive manufacturers to develop devices without the extra cost of shielding. Available in LLP-28 packaging, the device is priced at $0.85 in high volume quantities.

Industry and the Profession

Digital Over Power Line Standard Under Development

The ability to send high speed digital data over the power lines between substations and homes and offices is attracting increasing attention because it can make every wall outlet a portal to the Internet. In seeking to help realize this potential, the Institute of Electrical and Electronics Engineers (IEEE, **www.ieee.org**) has begun to develop IEEE P1675, "Standard for Broadband over Power Line Hardware."

When finished, IEEE P1675 will give electric utilities a comprehensive standard for installing the required hardware on distribution lines — both underground and overhead — which provide the infrastructure for broad-band-over-power-line (BPL) systems.

It will also include installation requirements for the protection of those who work on BPL equipment and ensure that such systems do not place the public at risk. The standard is targeted for completion in mid 2006.

Adding broadband capability to a local power distribution system is relatively straightforward. A computer-router combination and a coupler take the signal from an optical fiber cable as it enters a substation and imposes it on the electric current. The signal travels over the medium-voltage lines, with repeaters placed every 0.5 to 1 mile to keep the signal viable.

A repeater/router near a residence or business extracts the signal off the medium voltage just before the transformer and injects it onto the low voltage wiring on the other side of the transformer. The signal is now on all of the low voltage wiring within the structure and can be accessed at any outlet by plugging in a modem. Anyone from the utility, Internet service provider, and BPL equipment sectors who wants to help develop this standard is invited to join the IEEE 1675 Working Group. For more information on this standard and its working group, visit **http://grouper.ieee.org/groups/bop**

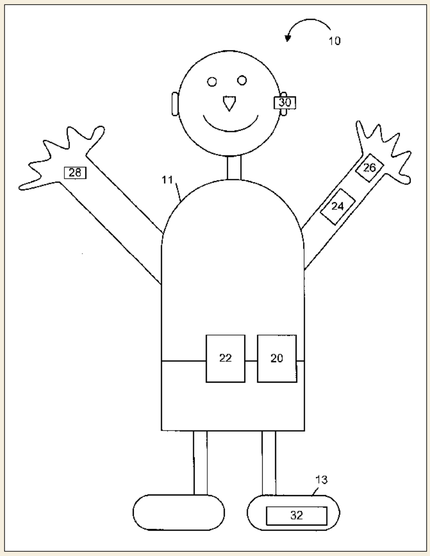
Dubious Achievement for US

Recent research at CipherTrust, Inc. (**ciphertrust.com**) — an Email security company — indicates that more than 80 percent of the Internet Protocol addresses sending spam are located in three geographic areas: Korea (28.58 percent), the US (28.41 percent), and China and Hong Kong (23.30 percent).

However, in terms of total volume, the US is clearly the king of all spammers, generating 85.93 percent, compared with 3.02 percent for Korea and 2.3 percent for China and Hong Kong. Only two other countries (Ukraine and Australia) accounted for more than 1 percent (1.47 and 1.24, respectively). **NV**

News Bytes

Grow a Thick Skin



Most of us in the programming industry aren't very impressed with the patents granted to Microsoft over the past few years, but three researchers recently struck gold with their idea. US Patent 6,754,472 was awarded to Williams, Vablais, and Bathiche for a, "method and apparatus for transmitting power and data using the human body." No, this isn't the dawn of the *Matrix*, but a novel way for wearable computers to move information between each other — without getting you tangled up in their wiring.

As the patent abstract explains, the human body is used as a conductive medium over which power and data is distributed. The inventors suggest using pulsed DC or AC, while differing frequencies could be used to selectively power different devices. Not to worry, the current moved is on the order of picoamps — much less that the shock from a doorknob on a winter day.

Imagine the future of wearable computing: Nike shoes convert small amounts of energy stolen from your walking gait to pump power through electrodes at your ankle. A small computer the size of a Band-Aid taped to your head behind your ear uses bone conduction to fill you in on traffic conditions during your drive home. Accelerometers in your watch measure the differential acceleration of your movements, and correlate that to a "stress index" — calling ahead to make sure that gin and tonic awaits you at home ...

Wash-n-Wear Electronics

While on the topic of wearable computing, it would be good to make note of the clever pressure sensors developed by Peratech, Ltd.

QTC — the novel material incorporated within Peratech's components — was discovered when a company co-founder was looking for a conducting glue and created a material that was an insulator under normal conditions (resistance in the order of 1,012 ohms), but turned into an effective "metal–like" conductor when pressure was applied (resistance drops to less than 1 ohm).

QTCs are novel in that — unlike normal electrical conductors — they exhibit conductance via quantum tunneling effects and have a resulting immense range of resistance when compressed, stretched, or twisted. The transition from insulator to conductor follows a smooth and repeatable curve with the resistance dropping exponentially.

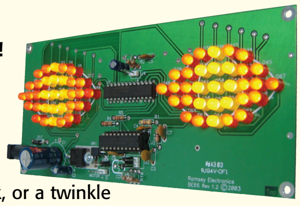
QTC Textile Sensors can be designed to interface with most electronic devices that are currently controlled or operated using switches or keyboards.

The field of wearable electronics has already grown in momentum, but

(continued on page 78)

RAMSEY
Spooktacular Halloween Treats!

Blinky-Eyes Animated Display



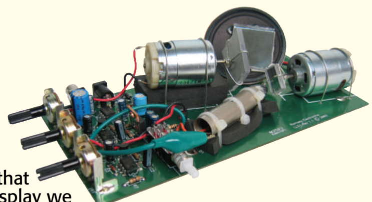
- ✔ **Animated display of 66 super bright LED's!**
- ✔ **Microcontroller controlled!**
- ✔ **Changes brightness automatically!**
- ✔ **Animated with constant motion!**

The ultimate animated LED kit that will dazzle you and delight your friends! Uses a microcontroller to randomly select from many different animations such as a long pause before a wink, or a twinkle of the eye to startle passers-by!

Four modes to satisfy any enthusiast's desires: 1. Off for long random periods, then blinks or winks. Designed to scare! 2. On for long periods before performing an animation, perfect for costumes and displays! 3. Animates all the time for constant motion, perfect display attention-getter. 4. Random fire! When placed in a pumpkin will light it up like you wouldn't believe! As if this weren't enough, the BE66 can also control a small hobby motor to shake bushes at random intervals or signal an external player to make a scary sound! Also has a CDS cell to sense light. In one mode, the display will dim as it gets dark for battery operation, and in the other it will turn off when it's too bright, so it plays only in the dark!

BE66	**Blinky-Eyes Animated Display Kit**	**$59.95**

Laser Light Show



- ✔ **Audio input modulates pattern!**
- ✔ **Adjustable pattern & size!**
- ✔ **Projects neat motorized patterns!**
- ✔ **Uses safe plastic mirrors!**

You've probably seen a laser show at concerts or on TV. They're pretty impressive to say the least! Knowing that you can't afford a professional laser display we challenged our engineers to design one that's neat and easy to build, yet inexpensive. Well, the result is the new LLS1 Laser Light Show! This thing is sweet and perfect for your haunted house or halloween parties! It utilizes two small motors and a small standard laser pointer as the basics. Then, we gave it variable pattern and speed controls to customize the pattern!

Not enough, you say? How about a line level audio input to modulate the pattern with your CD's, music, or spooky sound effects? You bet! Everything is included, even the small laser pointer. Runs on 6-12 VDC or our standard 12VDC AC Adapter (not included).

LLS1	**Laser Light Show Kit**	**$44.95**
PS21	**12VAC Output 110VAC Power Supply**	**$19.95**

High Intensity SMT Blinky Kit

- ✔ **Super bright SMT LED's!**
- ✔ **Subminiature board, perfect for hidden applications!**
- ✔ **Runs on miniature button cell batteries**

The BL2 is the perfect subminiature answer to high intensity flashing LED applications! Using SMT technology, the BL2 is small enough to conceal anywhere to provide alternating high intensity flashes! Need eyes for your mask? Just remote the LED's and you're all set! Install it on your kid's candy bucket as a neat attention grabbing display and a safety indicator! The applications are endless.

When complete, the BL2 provides alternating super bright red and blue LEDs. Runs on 6VDC, and we include two LR55 button cells and a cell holder to mount on the backside of the board. It's one complete unit! If you don't want SMT LED's you can also use regular T1¾ LED's (included). If you want to learn what SMT is all about, the BL2 is for you! We provide a detailed instruction and assembly manual, SMT theory information, and we even include spare SMT chips to cover you when you goof up! If you're interested in learning all about SMT technology and ending up with a really neat LED display, the BL2 is for you.

BL2	**High Intensity SMT Blinky Kit**	**$17.95**

Halloween Pumpkin

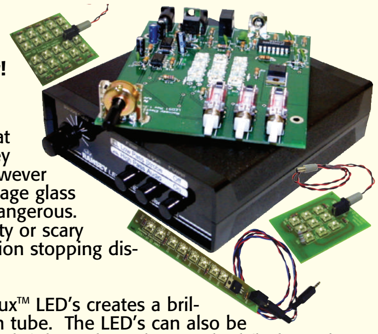
- ✔ **25 bright LED's!**
- ✔ **Random flash simulates flickering candle!**
- ✔ **Super bright LED illuminates entire pumpkin!**
- ✔ **Simple & safe 9V battery operation**

The perfect "starter" kit with a terrific Halloween theme! You won't be scraping the seeds and guts out of this pumpkin! Six transistor circuit provides a neat random flash pattern that looks just like a flickering candle. Then a super bright LED illuminates the entire pumpkin with a spooky glow!

The pumpkin face is the actual PC board, and assembly is easy through-hole soldering of all components and LED's. Your pumpkin is powered by a standard 9V battery (not included) which snaps to the back of the pumpkin. An on/off switch is also included. Create a new kind of pumpkin this year, and learn about LED's and electronics at the same time!

MK145	**Electronic Halloween Pumpkin Kit**	**$9.95**

LED Strobe Light



- ✔ **Everlasting LED's won't burn out!**
- ✔ **Variable flash rate & audio trigger!**
- ✔ **Bass & treble trigger modes!**
- ✔ **Safe low voltage operation!**

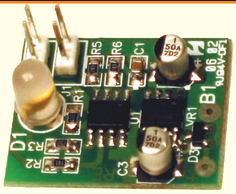
You have all seen those strobe lights at the special gift shops in the mall. They have been around for a long time, however they all feature one thing... a high voltage glass Xenon tube, that is both fragile and dangerous. Now you can illuminate your next party or scary Halloween setting with the same motion stopping display without worry!

A plug-in 3x3 array of super bright Telux™ LED's creates a brilliant sharp flash just like a Xenon flash tube. The LED's can also be mounted directly on the main PC Board if desired. In the standard flash mode, a variable rate control varies the flash frequency from approx 1 to 220 flashes per second. In the audio sync mode, the flash is triggered by any audio input you provide into the standard RCA audio input connector. Built-in low and high pass filters allow you to select either bass or treble music triggering! Just picture the strobe reacting to those Halloween sound effects CD's! An external trigger in/out connector lets you connect multiple units together for simultaneous flash.

Optional plug-in display boards with 8 or 20 LED's are available for even more strobing power! These are perfect to put inside a pumpkin! Runs on 12-15 VDC, and unlike the high voltage Xenon strobes of the past, the LEDS1 is safe for the kids to build! Includes a matching custom case and knob set to give your strobe light a great finished look. The plug-in display boards may be mounted on top of the case, or remotely located

LEDS1C	**High Power LED Strobe Light Kit With Case**	**$44.95**
LEDS8	**Display Board, Inline Array of 8 LED's**	**$17.95**
LEDS20	**Display Board, 5x4 Array of 20 LED's**	**$29.95**
AC125	**110VAC Power Supply**	**$9.95**

Multi-Color SMT Blinky Kit



- ✔ **Bright full color LED - red, green blue elements!**
- ✔ **8-pin microcontroller!**
- ✔ **Operates on 6 VDC to 12 VDC**
- ✔ **Extra SMT components are included!**

If you're looking to put some stunning color into your Halloween displays, this is the kit for you! It was originally designed as a way to show off the latest technology in RGB (Red, Green, Blue) full color LEDs while providing kit builders with a fun and economical SMT practice kit.

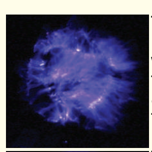
Uses PWM methods to generate any color with a simple 8-pin microcontroller, with switchable speed selection as well! The dual jumper control system tells the microprocessor what scanning speed is desired for a full cycle of 16,777,216 possible colors from the LED! It randomly selects different color scanning modes to wow any observer to the point of distraction! This little attention getter is fun to build and has loads of possible applications. To say it is an attention getter is an understatement! Just wait till you see the LED smoothly change colors in front of your eyes!

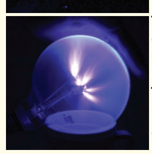
Operates from 6VDC to 12VDC, and will run continuously on a standard 9V battery for several days! And if you're new to SMT technology don't worry, we include extra SMT components to cover your mistakes!

SBRGB1	**Multi-Color SMT Blinky Kit**	**$29.95**

Plasma Generator

- ✔ **Generates 2" sparks to a handheld screwdriver!**
- ✔ **Light fluorescent tubes without wires!**
- ✔ **Build your own plasma balls!**
- ✔ **Generate up to 25kV @ 20KHz from a solid state circuit!**





This new kit was conceived by one of our engineers who likes to play with things that can generate large, loud sparks, and other frightening devices. During the process of looking for parts for one of his latest experiments he discovered how difficult it was to find a high voltage transformer that met his requirements. Well, we had a super unit designed expressly for us! The result... the PG13 Plasma Generator designed to provide a startling display of high voltage!

This is one of the neatest Halloween kits around. It really serves no purpose other than producing stunning lighting displays, drawing big sparks, scaring the neighbors and performing lots of high voltage experiments. In the picture, we took a regular clear "Decora" style light bulb and connected it to the PG13 - WOW! A storm of sparks, light tracers and plasma filled the bulb. Holding your hand on the bulb doesn't hurt a bit and you can control the discharge! It can also be used for powering other experiments; let your imagination be your guide! Can also be run from 5-24VDC so the output voltage can be directly adjusted.

PG13	**Plasma Generator Kit**	**$64.95**
PS21	**110VAC input, 16VAC output power supply**	**$19.95**

Putting the Spotlight on BASIC Stamp Projects, Hints, and Tips

Stamp Applications

Measuring Up — Up to 80 Centimeters, That Is

Add a bit of intelligence to your Halloween displays with IR distance measuring.

The night is drawing closer ... my favorite night of the whole year: Halloween. I love Halloween — the costumes, haunted houses, parties, and friendly exchanges with trick-or-treaters; Halloween is the best. When I have the chance, something I like to do is build Halloween-oriented props and decorations and you can bet that many of those props get some sort of automation via the BASIC Stamp microcontroller.

Good Halloween props add an element of surprise, which, of course, intensifies the fright — and that's the most fun thing about Halloween, right? The only problem is that, as a society, we are far more sophisticated than we were in the past (especially the teenagers). We can easily see through a cheesy effect and find the trigger, which ruins the effect for those who immediately follow.

Instead of using a fixed-point trigger for an automated prop, what if we used a distance measuring device so that we could select a random trigger point? That would keep 'em guessing, wouldn't it? You bet! We've used sonic measuring devices in the past (SRF-04 and SRF-08); this time, we'll do it with infrared. The device we're going to use is the low cost Sharp GP2D12.

Read Volts, Get Distance

There is no great mystery to using the GP2D12: We simply connect it to an appropriate analog-to-digital converter and read the output voltage. The voltage is then converted to distance.

The first part is very easy. For this project, we'll use the ADC0831 analog-to-digital converter — a part we've used before and should have no trouble with. In order to simplify the project code, we'll connect the wiper of a mutli-turn pot to the Vref input of the ADC0831 and set this to 2.55 volts. What this does for us is set each output count to be equal to 0.01 volts (255 [max count] divided by 2.55 [Vref] = 0.01 volts/count). Figure 1 shows the schematic for the project.

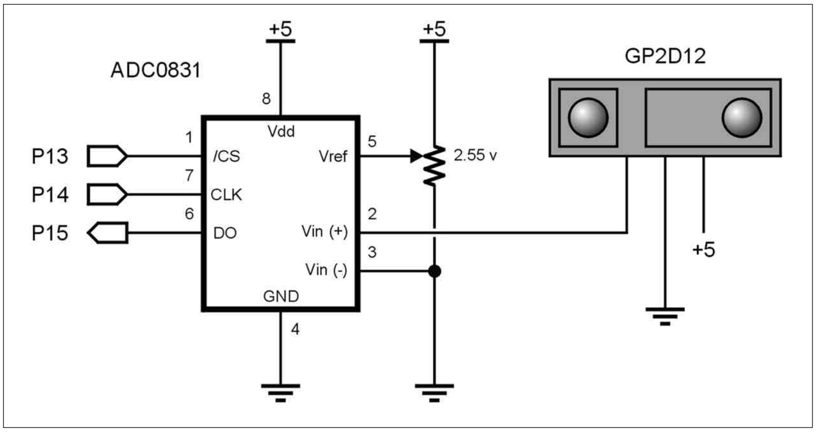
Let's have a look at the code that reads the voltage from the ADC0831:

```
Read_0831:
  LOW AdcCS
  SHIFTIN AdcDta, AdcClk, MSBPOST, [result\9]
  HIGH AdcCS
  RETURN
```

This code is straightforward, but — if you haven't used the ADC0831 before — you may be wondering why we need nine clocks for an eight-bit value. As always, you should download the documentation for any part you're working with and, when you look at the ADC0831 timing chart, you'll see that the ADC conversion is started by bringing the CS (chip select) line low, then putting a pulse on the clock line. Here's where we get the extra clock pulse. The value bits are clocked out, MSB to LSB, with the following eight clock pulses. Each ADC bit is valid after the falling edge of the clock, so we use MSBPOST to read the bits. Once all the bits are clocked in, the device is deselected by bringing the CS line back high.

Okay, that's done, but what we're likely to run into is a bit of jitter in actual application. An easy way to smooth this jitter is to take the average of multiple readings. Let's do it:

Figure 1. ADC0831/GP2D12 connections.



```
Read_GP2D12:
  cVolts = 0
  FOR idx = 1 TO 3
    GOSUB Read_0831
    cVolts = cVolts + result
    PAUSE 30
  NEXT
  cVolts = cVolts / 3
  RETURN
```

We start by clearing the old cVolts value, then, with a loop, take three readings of the ADC0831 and accumulate them. Keep in mind that we will need to use a Word-sized variable for cVolts, otherwise we'd likely get a roll-over error after the second reading. At the end of the loop, we divide the accumulation by the number of loop iterations to get the average value.

What happens, though, when we're in a pinch for variable space? One way around this — though likely to be slightly less accurate than the method above — is to divide each reading before accumulating. Keep in mind that the lower readings and larger divisors result in a greater likelihood for error. If you keep the divisor small, this shouldn't become too much of a problem. Here's the code for the alternate version:

```
Read_GP2D12_Alternate:
  cVolts = 0
  FOR idx = 1 TO 3
    GOSUB Read_0831
    cVolts = cVolts + (result / 3)
    PAUSE 30
  NEXT
  RETURN
```

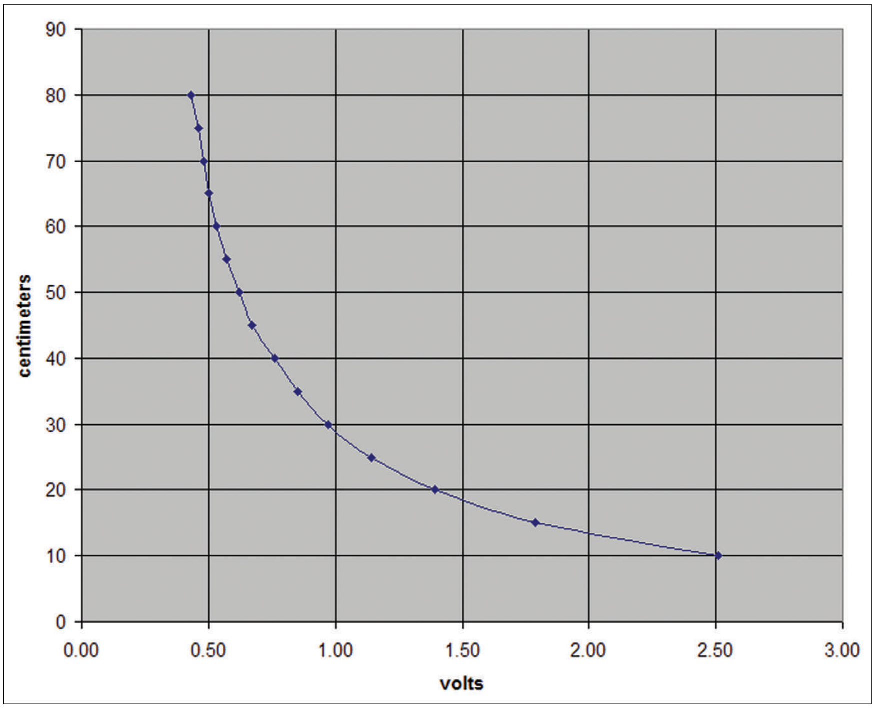


Figure 2. GP2D12 output voltage versus distance.

Straightening the Curve

Now comes the tricky part — converting the voltage output of the GP2D12 to a distance value. Have a look at Figure 2 and you'll see why I say this is tricky. Over the entire measurement range, the output from GP2D12 is not at all linear in respect to distance, so a simple $mx + b$ equation is just not going to work. I plugged the data into a curve fitting program and found that it takes a fourth-order equation to get anywhere close to the data set.

Applying a fourth-order equation with 16-bit integer-only math is just not very practical.

There are interesting solutions to this dilemma, but most of them were more than I wanted to wrap my brain around, so I decided simple is better than interesting. (My middle name, after all, is "Simple.") Looking at the graph again, we can see that the segments between data points are not far from the curve that would fit between those same points. What I decided to do, then, is to calculate the slope between data points and interpolate from there. I felt like this was an acceptable solution, given the slightly loose specifications of the GP2D12. (It is a low cost device.)

First things first — that curve in Figure 2 actually came from my sensor, bounced off an 18% gray card (something photographers use). Using some cardboard and foam blocks, I set up and marked a test jig at 5 centimeter intervals, then measured the voltage at each interval from 10 to 80 centimeters using the code we've developed thus far.

Now what? As I just mentioned, the segments between data points can be treated as a line, so what we can do is find the data points that surround our current reading, calculate the slope of the line between them, and then interpolate the distance. Let's have a look at the code that does this, then work our way through.

Here's the table of distance readings from my test setup:

```
Vout           DATA    251, 179, 139, 114,  97
               DATA     85,  76,  67,  62,  57
               DATA     53,  50,  48,  46,  43
               DATA      0
```

Now, here's the code that uses the table and the current voltage reading:

```
Estimate_Cm:
  FOR idx = 0 TO 15
    READ (Vout + idx), test2
    IF (test2 <= cVolts) THEN EXIT
  NEXT

  SELECT idx
    CASE 0
      cm = 10

    CASE 1 TO 14
      cm = 10 + (5 * idx)
      IF (test2 <> cVolts) THEN
        READ (Vout + idx - 1), test1
        slope = (test1 - test2) * 10 / Xspan
        cm = cm - ((cvolts - test2) * 10 / slope)
      ENDIF

    CASE 15
      cm = 80
  ENDSELECT
  RETURN
```

The first part of the process is locating the position of the current reading vis-à-vis the table values from our test setup. Since the table is very small, the simplest method is to loop through the possible values until we find the test point that is less than or equal to our current voltage reading. We can use EXIT to terminate the loop early when we find a match.

On the extremes — when idx is either 0 or 15 — we simply set the distance reading to the minimum or maximum values. When I first started working with the code, I tried to provide an "out of range" calculation, but — the way the output falls on the data points — this just didn't work out very well. So, keep this in mind when using the GP2D12 with this code: A reading of 10 cm actually means 10 centimeters or less and a reading of 80 cm means 80 centimeters or greater.

Things get interesting when idx is between 1 and 14. The first step is to calculate the rough distance using idx. Next, we check to see if the value of test2 is not equal to cVolts, because, if it is, we're done and have the distance value in hand. Most of the time, test2 will be less than cVolts, so we'll find the other value that borders (is greater than) our current reading and interpolate from there.

At this point, we already have the table value lower than cVolts; what we do next is subtract one from idx and read the value that is greater than cVolts. We'll put this

value in test1. Now that we have the table values surrounding our input from the GP2D12, we can calculate the slope between them by taking the difference and dividing by the span between these points (5 centimeters in our test data). Since we're doing division and the values on the outer end of the range get very small, we'll multiply the difference by 10 before dividing. This will prevent getting a slope value of 0.

We're almost done. The final step is to divide the difference between our current reading (cVolts) and test2 by the slope, then subtract that from the rough calculation of distance. Again, we'll multiply the difference value by 10 — this time to remove the offset introduced by the way we calculated the slope.

Just to make things crystal clear, let's work through a set of numbers. We'll start with an input voltage of 2.10 volts. The table search will set idx to 1, as this entry (179) is the first value, less the current value of cVolts. Our rough calculation of distance, then, is 15 centimeters. At this point, test2 is indeed less than cVolts, so we have to read the next lower table value (251) and place this into test1. Using 251 and 179 for test1 and test2, we get a slope value of 144; at this point, slope is in millivolts per cm. Using the BASIC Stamp's integer math, the difference from our rough distance calculation works out like this:

((210 – 179) * 10 / 144 = 2

When we subtract 2 from our rough calculation, we end up with a distance reading of 13 centimeters.

Okay, so much for the theory, how does it work in practice? I marked up my test rig at one centimeter intervals and found that it worked pretty well; the readings across the range were within a centimeter of the actual distance to my target. I found this perfectly acceptable, given the (slightly loose) specifications of the GP2D12.

The reason I developed the code I did is that it's very easy to plug in different sensor values. I elected to use a **DATA** table instead of **LOOKUP** so that the program can be more easily expanded with more table entries. (**LOOKUP** tables beyond a few values can get unwieldy.) If you'd like to find a way to plug the voltage value into a formula in order to get the distance value, I encourage you to visit the Acroname website and look at their application note on the GP2D12. That note goes into a very detailed discussion of finding slope and offset points to linearize the output from the GP2D12. It's a little bit complicated and requires some experimentation, but you may find this method valuable.

Scare 'Em, Danno

Before we head out, let's chat a bit about using the sensor as I suggested at the beginning of the article. As I've frequently mentioned in the past, we can learn a lot by mimicking what pros have already done. I was in a public washroom a few days ago and the sinks had automated faucets. When one places one's hands about six inches from the nozzle, the water starts running.

How would you program the BASIC Stamp to mimic the faucet control (to apply it to a Halloween display)? This would be my strategy:

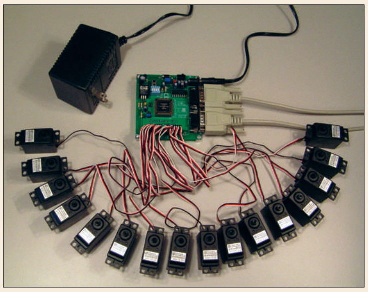
1. Measure distance to target.
2. Is distance less than threshold?
3. If no, go back to Step 1.
4. If yes, check several more times with a delay in between.
5. If target stays in range, trigger the device.
6. Add a [random] delay, allow the prop to run, and reset.
7. Go back to Step 1.

Can you do it? Of course you can — you're a BASIC Stamp programmer!

Have a safe and happy Halloween. Until next time, Happy Stamping. **NV**

Resources
Jon Williams
jwilliams@parallax.com
Parallax, Inc.
www.parallax.com

SERVOCENTER 3.1: MODULE ALLOWS UNPRECEDENTED SPEED AND POSITION CONTROL



Yost Engineering, Inc., has introduced ServoCenter 3.1, an embedded R/C servo motor controller allowing unprecedented independent control of both speed and positioning for up to 16 servos per board and 16 daisy-chained boards. Using only one serial port, unique speed and positioning parameters can be passed to each of 256 motors.

Unlike other controllers, this independent control of servo position and speed makes ServoCenter especially useful for applications such as robotics, animatronics, motion control, automation, retail displays, and other areas where independent, coordinated, fluid motion is desired. A scaled positioning mode makes it easy to set maximum, minimum, and startup points and the speed control feature allows each servo to seek at a rate from 1% to 100% of its full speed.

Example programs are provided in GCC/Linux, QBASIC, VB.NET, C#.NET. VC.NET, VB 6.0, VC++ 6.0, and Turbo C for both simple raw serial protocol and the included ActiveX control and DLL. An onboard regulator provides 6.0 V or 4.8 V with over current, thermal protection, selectable baud rate, and flexible power options including battery usage.

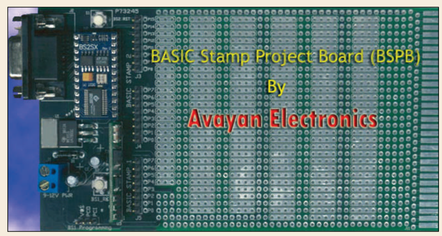
A complete package of ServoCenter, nine-pin serial cable, AC adapter, user's manual and programming guide, and software/examples CD is $69.95 (ServoCenter board only for $48.95).

For more information, contact:

YOST ENGINEERING, INC.
630 Second St.
Portsmouth, OH 45662
Tel: **888-395-9029**
Email: **sales@YostEngineering.com**
Web:
www.YostEngineering.com/ServoCenter

Circle #38 on the Reader Service Card.

PROTOTYPING TOOLS FOR BASIC STAMP USERS



Avayan Electronics now offers a powerful BASIC Stamp project development board — the BSPB. This board allows the user to interface BASIC Stamp I and BASIC Stamp II compatible modules at the same time to the devices pertaining to the application at hand.

The BSPB offers access to all ports in an "easy to get to" fashion by providing solder pads, as well as pin stakes. Both modules have an individual reset switch and their respective serial programming port. Power is regulated in the board at the same time it is distributed through the prototyping space for ease of use.

Although the prototyping space will be more than enough for most applications, running out of space is not an issue. An edge card connector at the end of the board allows

expansion with their Project Board Mother Board (PBMB) and the additional Prototype Expander Boards (ProtoXP).

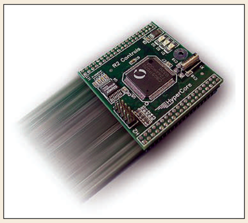
The board can be purchased through Avayan Electronics for $29.95.

For more information, contact:

AVAYAN ELECTRONICS
PO Box 994
Webster, NY 14580
Tel: **585-305-5430**
Web: **www.avayanelectronics.com**

Circle #79 on the Reader Service Card.

MICROPROCESSOR CORE MODULE



R2 Controls announced the general availability of the R108 Hypercore Microprocessor Core Module. The Hypercore Microprocessor Core Module is designed to simplify integration and engineering of embedded microcontroller projects and thereby reduce the time to market and increase profitability.

Based on the Silicon Laboratories™ 8051 system-on-chip processor, the Hypercore combines 64K of Flash, 128K of SRAM, and amazingly fast performance with a vast array of I/O, serial communications capabilities, and connectors to enhance integration with most any controller project design.

The R108 Hypercore Microprocessor Core Module is designed for engineers who need more than a simple, stripped down core module. The R2 Controls Hypercore comes standard with up to 16 precision analog channels with up to 12-bit resolution. Two 12-bit analog outputs are available. The Hypercore is also readily adaptable into the production platforms of many projects because of its extremely small footprint. It is low cost — in comparison with engineering its capabilities into a larger or more complex controller — and has a high performance of the 25 MIPS microprocessor.

Specifications and Performance Highlights

The R108 Hypercore Microprocessor Core Module incorporates the Silicon Laboratories C8051F020 microprocessor running at 22.1 MHz with pipelined instruction architecture executing 70% of the instructions in one to three system clocks. There are 64K of Flash memory and 128K of SRAM with battery backup. The Hypercore includes eight 12-bit high precision resolution and eight 8-bit resolution analog inputs, as well as two12-bit analog outputs. There are over 30 digital I/Os.

Communications capabilities include two serial ports:one RS232 and one TTL. SPITM and I2CTM are also supported. A JTAG port is provided for connecting to a PC for programming in C. Basic programming is accomplished through the serial port. Five 16-bit timers are available. The Hypercore operates at 5-9 VDC and consumes less than .25 watts. Power monitoring and sleep mode functions are also supported.

R2 Controls provides a choice of two Hypercore Module Development Kits to facilitate the rapid and effective utilization of the R108 Hypercore Microprocessor Core Module. The “Basic” Hypercore Development Kit is used for programming in BASIC and includes the development board, the R108 Hypercore Core Module, the power transformer, a serial cable and a CD with the Basic Interpreter Software, the API, the operating instructions, and sample programs. The “Deluxe” Development Kit contains all of these components, plus a programming

adapter and cable, which enables the R108 Hypercore to be programmed in C.

"The R108 Hypercore Microprocessor Core Module will integrate with almost any industrial microcontroller product on the market today," states R2 Controls Chief Technology Officer Rich Kirkpatrick. R2 Controls also provides telephone technical support to aid customers during project development and deployment.

Price and Availability — Introductory Offer

The Basic Hypercore Development Kit is priced at $169.00. The Deluxe Hypercore Development Kit is priced at $229.00. The R108 Hypercore Microprocessor Core Module is priced at $79.00 (single quantity).

For more information, contact:

R2 CONTROLS
Tel: **866-797-9229**
Email: **Sales@R2-controls.com**
Web: **www.R2-controls.com**

Circle #61 on the Reader Service Card.

XGAMESTATION MICRO EDITION

The XGameStation Micro Edition — "ME" — is the world's first video game system development kit designed for education. The kit comes with an assembled XGameStation console, a single controller, all necessary cables, a CD containing all system software and tools necessary to develop for the system, and — of course — an extensive eBook that explains how the system was designed and how it works from the ground up. Everything from basic digital logic to computer engineering to circuit board design to firmware and low-level software is covered in precise detail. The XGS Micro Edition is powered by the Ubicom SX52 running at 80 MHz for 80 MIPS of performance with a 12.5 ns instruction cycle.

XGS System Architecture

The XGS Micro Edition's hardware was inspired by retro designs — such as the Atari 2600, Atari 800, c64, and Apple II — to give the system the most flexibility possible while, at the same time, keeping the hardware complexity to a minimum so users can understand the system. Therefore, the graphics are generated via a software/hardware combination without a dedicated frame buffer or sprite system.

The SX52 core is responsible for general control of the raster timing and video signals generated via controlling a special D/A converter that generates a TV level signal and mixes Luma and Chroma to generate composite video. The SX52 generates the sync pulses, as well as the raster data.

Sound on the XGS Micro Edition is generated with a ROHM BU8763 — a three-channel FM synthesizer with full envelope control. For I/O, the XGS has two Atari 2600 compatible joystick ports, a serial port, and a 30-pin expansion port. Rounding out the hardware is an external 128Kx8 SRAM for general use, such as program data, decompression buffers, screen buffers, and more. Of course, the XGS ME comes with a built-in programmer and tool chain — XGS Studio. Additionally, the XGS has a compatibility port for Parallax, Inc.'s SX-KEY, so users can use their tools, as well.

For more information, contact:

NURVE NETWORKS LLC
402 Camino Arroyo West
Danville, CA 94506
925-736-2098 Fax: **925-736-2128**
Email: **support@nurve.net**
Web: **www.xgamestation.com**

Circle #75 on the Reader Service Card.

UNIVERSAL INPUT LED ACID CHARGER

Cell-Con, Inc., announces the availability of a new line of cost effective universal input lead acid chargers.

The input is 90-264 VAC/47-63 Hz with output voltages of 6, 12, 24 standard up to 10 amps. Safety approvals include UL and CE. It contains an integral LED status indicator.

This product is available for immediate delivery. Depending upon exact model, single piece pricing is under $65.00.

This product can be viewed at the Cell-Con website.

For more information, contact:

CELL-CON, INC.
Tel: **800-771-7139 ext. 210**
Web: **www.cell-con.com**

Circle #29 on the Reader Service Card.

Circle #31 on the Reader Service Card.

Electronics Q&A

In this column, I answer questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist.

Feel free to participate with your questions, as well as comments and suggestions.

You can reach me at:
TJBYERS@aol.com

What's Up:

Converting this to that is the theme: USB, SCSI, and PC sound. A few simple circuits for minor applications and optocouplers demystified. Need a Word template for Power of Attorney or a will? Got you covered.

Relay Pulser

Q. I'd be interested in a circuit that pulses a stepper coil every five to six seconds, using its own contacts and as few components as possible, maybe just an R/C combination. I have steppers with a multitude of accessory contacts. The circuit could work by charging C to the trip point of the relay, but wouldn't some path for discharge also be required?

Phillip Milks
via Internet

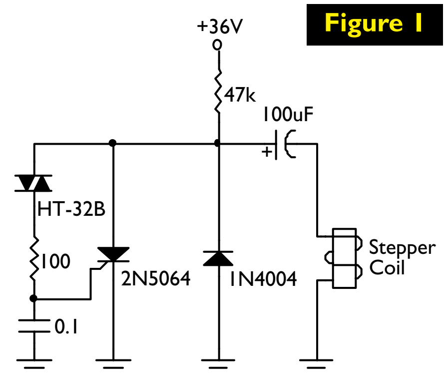
A. I am assuming that what you have are older rotary steppers that were used by Ma Bell years ago (because rotary relays aren't sold today) to count the pulses of a telephone dialer. As a kid, I used to play with war surplus rotary switches (called impulse relays) that I was able to wire for progressive triggering. However, without the five second pause — which I could never achieve — it sounded like a machine gun. The contacts just weren't in the right number or order. So, I doubt you'll be able to pull this off without an external trigger.

You can always use a 555 astable oscillator, but it's overkill for this application. What I'd do is use an R/C timer that triggers an SCR to advance the stepper. In the design in Figure 1, I adopt a circuit that's more commonly used with CDI — capacitance discharge ignition systems. Instead of using "points" to trigger the coil, I'm using a diac.

A diac is a breakover diode that conducts current when the voltage exceeds a threshold value. In this case, the breakover voltage is between 30 and 34 volts. As the 100 µF cap charges through the 47K resistor, the voltage will eventually exceed the breakover voltage, causing the SCR to turn on and discharge the cap through the stepper coil, thus advancing your stepper.

The 100 µF and 47K values are selected for about a five second pulse time; you can change this time using the formula t = RC. Did I forget to mention relay coil voltage? Who cares! Coils are current-operated and the jolt this circuit provides will actuate any relay up to 36 volts without damage. It is the in-rush current that engages the coil, not the voltage.

Figure 1



5-sec Coil Pulser

Simple Tach

Q. Years ago, I built an adapter to measure the dwell angle of my car (the time the points are closed on a gas engine) using an analog multimeter. Have you ever done a story on building an adapter to display engine RPM using a DMM?

Dennis
via Internet

A. No, but it's easy enough to do. The concept is to trigger off the closing of the points and translate that into a voltage that can be read by a voltmeter — including a DMM. The more frequent the trigger, the faster the RPM, the higher the voltage; it's basically

a frequency-to-voltage converter. The circuit in Figure 2 does just that.

It uses a 555 monostable timer to output a defined-width pulse every time the trigger input (pin 2) is activated. The output pulse train is then averaged to produce a "steady" DC output. Using the CAL pot, the width of the pulse is adjusted so that a 5,000 RPM input produces a 5 volt output.

Be warned, though, that most cheap DMMs won't respond to instantaneous changes in engine speed. That's because they have a low sampling rate, so the reading will lag behind the actual RPM by a second or two while the DMM display stabilizes. An analog meter (one with a pointer) will respond faster. A 5K resistor in series with a 0-1 mA panel meter will show 5,000 RPM at full scale.

PS/2 to USB

Q. Is there any way — perhaps with hardware — to turn a regular serial mouse into a USB mouse? I am interested in the data communication aspect.

Sassan
via Internet

A. It depends on what you consider to be a "regular" serial mouse. Mice come in a number of shapes, sizes, and interfaces. Just a few years ago, serial mice (with DB-9 connectors) were quite popular, but the computer industry has all but abandoned them in support of USB and PS/2 devices. Each interface uses a different protocol and herein lies the problem.

The easiest conversion is between a USB "A" female to PS/2 male adapter, which lets you plug a USB mouse into an PS/2 mouse port. However, this will not work with every USB mouse! Your USB mouse must be made so that it can use one of these adapters.

Going from a PS/2 mouse to a USB port takes a lot more work. That's because the PS/2 protocol uses interrupts for its serial communications and the USB doesn't. This requires both hardware and software. A favorite device among hobbyists is the dual-port adapter that lets you plug both your PS/2 mouse and keyboard into one USB port. USB adapters run between $12.00 and $75.00 — depending on features and compatibility — and are available from RadioShack (26-226) and most computer stores. Other conversion protocols (parallel to USB, etc.) are also available.

For those readers who wish to make it yourself, Microchip serves up a "PS/2 to USB Mouse Translator" application note (TB055) that uses a 16C745 microcontroller (**ww1.microchip.com/downloads/en/AppNotes/91055C.pdf**). A download of the software is available from Microchip; you have to program the chip yourself.

IDE to SCSI?

Q. This question is about SCSI and IDE hard drives. I recently acquired a new SCSI hard drive and would like to install it into my PC, which has a built-in IDE controller. Is

Gas Engine Tach

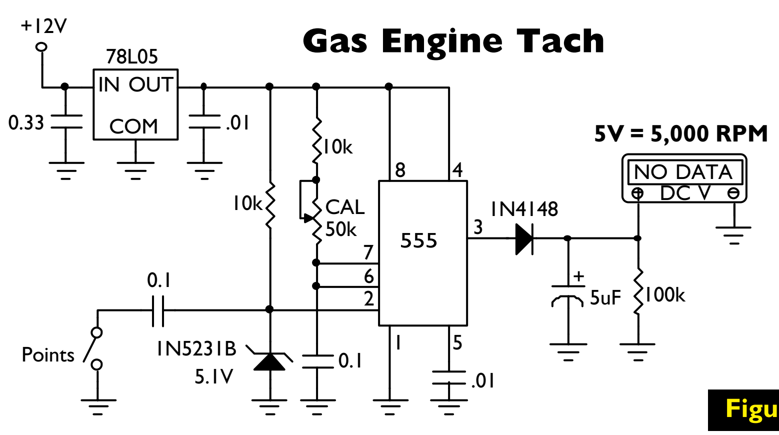


Figure 2

Order Mark	Part Number	Fig.	Retail Price (Volume Discount Available)	Emitted LED Color	Epoxy Lens Color	Chip Material	MAX	I_f (mA) typ	I_v (mcd) typ	V_f (V) typ/max	λ_p (nm)	$\Delta\lambda$ (nm)	Full Viewing Angle ($2\theta_{½}$)
High-Efficiency Diffused Lens LEDs													
	L185TR1	A		Red	Red Dif	GaAlAs/GaAlAs		10	0.5	1.7/2.0	655	24	64°
	L185TR3	B		HE Red	Red Dif	GaAsP/GaP		20	24	2.0/2.6	635	45	65°
	L185TY1	B		Yellow	Yel Dif	GaAsP/GaP		20	20	2.1/2.6	585	35	65°
	L185TG1	B		HE Green	Grn Dif	GaP/GaP		20	30	2.2/2.6	568	30	40°
High-Efficiency Clear-Tinted Lens LEDs													
	L185CR4	B		HE Red	Red Clr	GaAsP/GaP		20	60	2.0/2.6	635	20	30°
	L185CO4	B		Orange	Orng Clr	InGaAlP		20	40	2.0/2.6	610	35	30°
	L185CY4	B		Yellow	Yel Clr	GaAsP/GaP		20	50	2.1/2.6	585	35	30°
	L185CG4B	B		Aqua Grn	Grn Clr	GaP/GaP		20	60	2.0/2.6	568	30	30°
Water-Clear Lens LEDs													
	L190CWR3KF-50D	B		Ultra Red	Water-Clr	GaAlAs/GaAlAs		20	800	2.0/2.6	660	20	45°
	L190CWE3KF-50D	B		Super Red	Water-Clr	InGaAlP		20	1800	2.2/2.4	630	17	45°
	L190CWO3KF-50D	B		Spr Orange	Water-Clr	InGaAlP		20	1200	2.2/2.6	611	20	45°
	L190CWY3KF-50D	B		Spr Yellow	Water-Clr	GaAsP/GaP		20	2500	2.2/2.6	595	15	30°
	L190CWG1K-50D	B		Super Green	Water-Clr	InGaAlP		20	560	2.0/2.6	570	15	45°

Ledtronics LED Lens Options

Figure 3

there an adapter that will convert the 40-pin IDE connector into a 50-pin SCSI plug? I know there are controllers out there, but they are so expensive. Is there something I can build myself?

George C. Boone, II
Radford, VA

A. IDE hard drives are so popular because they are cheaper than SCSI, which explains why you have an IDE controller built into your motherboard, rather than an SCSI. Unfortunately, you can't interface the IDE to SCSI using a simple adapter cable because — like the USB question above — the interfaces use different protocols. That is, they don't speak the same language. Not only are the words different, but so is the syntax. When one says, "Push," the other says, "Huh?"

Your best bet is to buy a SCSI controller board and plug it into an empty ISA slot. They can be found for under $50.00 at most computer stores and Jameco (800-831-4242; **www.jameco.com**). Moreover, the SCSI controller can manage up to 15 devices as opposed to the four-drive limit of IDE.

My suggestion? Stay with IDE and leave SCSI for the high-end boys.

LED Lens Evolution

Q. I have an LED that has a smoky black lens, but it doesn't emit light when I apply power. Is it an infrared LED or what? I thought IR LEDs were clear.

Ralph D.
via Internet

A. If memory serves me — and in this case, it does — my first LEDs were IR and they were smoky to opaque black. They also had very little output in the IR spectrum. That had nothing to do with the black lens, but rather the very low efficiency of the LEDs in those days.

The color of today's lenses is an evolution of increased output power, viewing angle, and larger color choice.

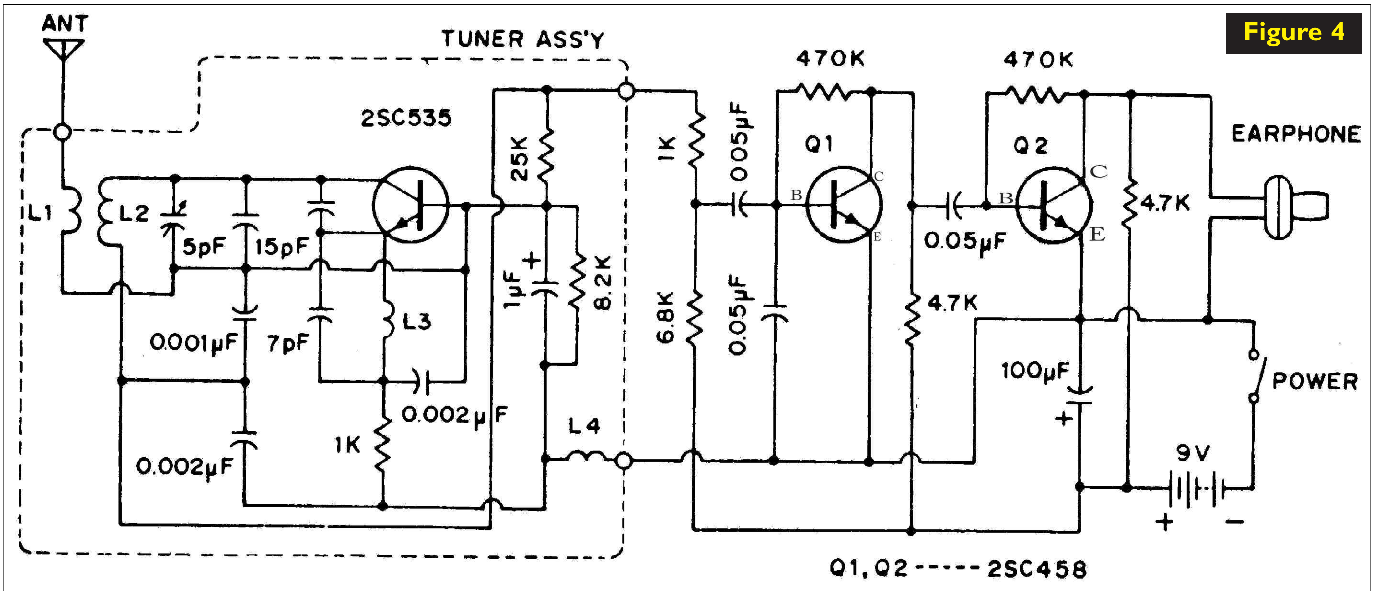


Figure 4

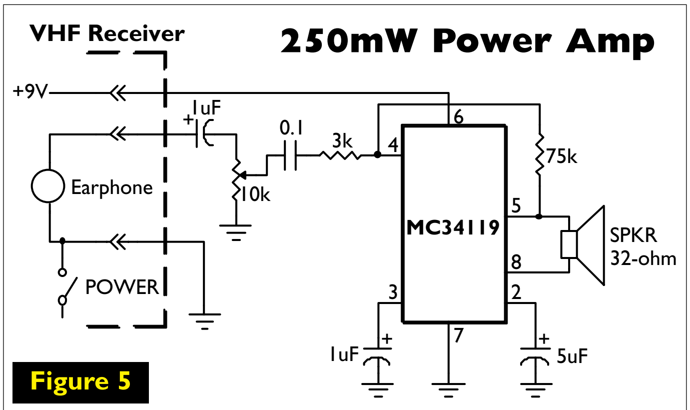


Figure 5

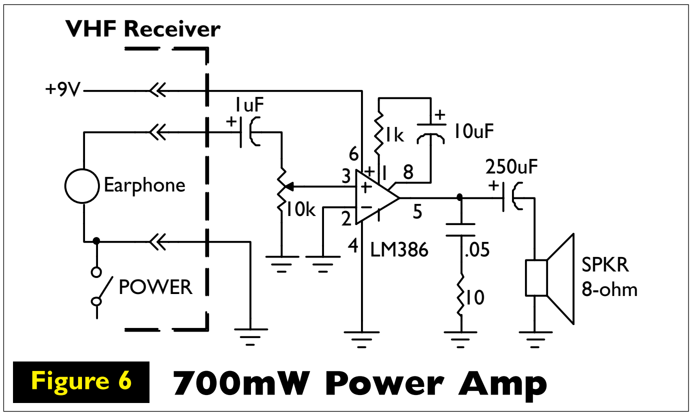


Figure 6 700mW Power Amp

Colored lenses fall into two categories: diffused and clear tinted. Diffused LEDs are most often used as indicators on PC boards, where a wider viewing angle is more important than illumination or color purity. Clear tinted LEDs are largely found in instrument panels, where they display on/off status, operating mode, and other visual information. Water clear LEDs have the most output power, highest color purity, and strong IR emission. The chart in Figure 3 shows a comparison of the different LED lens types.

Power Amp for Earphone Radio

Q. I would like an audio amplifier to replace the earphone for a battery-powered VHF receiver I have (Figure 4). I'd like it to provide a LOUD output from an 8 Ω speaker. I don't want a Walkman amp because they use external batteries. I want the amp to be powered by the internal battery of the receiver. I have a Motorola MC34119 chip that I'd like to use as the amplifier. Is it a good match?

Mike
via Internet

A. Yes, the MC34119 is a perfect match for this portable radio because it draws just 4 mA of quiescent current (volume all the way down) and about 100 µA when shut down. (Pin 1 can also be used for a squelch control.) The component count is a sparse six — and it needs no output capacitor! I don't think you're going to get the loud volume you're expecting, though. At 9 volts, the output power will be about 1/4 watt. That's loud for a radio of its size, but hardly a boombox. Also, you need a 32 Ω speaker to get that. With that said, Figure 5 shows the schematic you want.

For louder sound, I recommend the LM386, which you'll find in Figure 6. It, too, has a quiescent current of 4 mA, yet puts out three times the wattage. It does that using an 8 Ω speaker. The downside is that, the louder you play the radio, the faster you use up your battery.

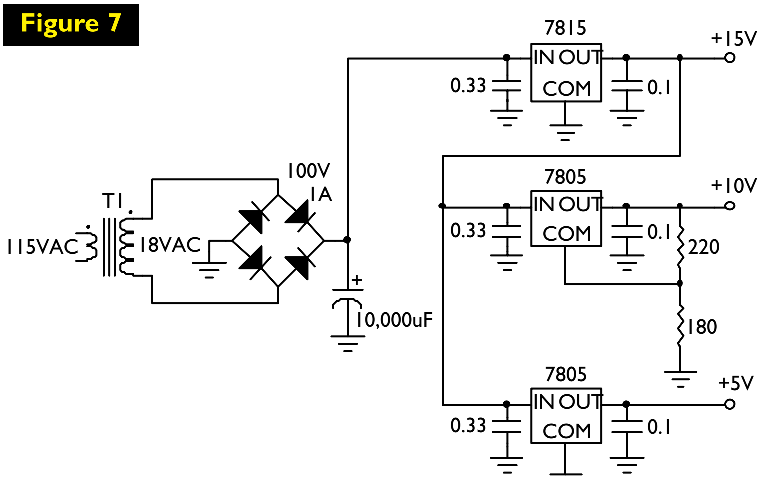
LCD to VGA?

Q. Is there a way to attach a laptop LCD to a PC via a video board? I have an old laptop and I'm wondering if I should try to sell it on eBay, donate it, or throw it away. Recently, I've wondered if, instead, I should save the LCDs and use them as monitors for other systems — like a clone with SVGA or VGA video adapter. Can this be done without costing me an arm and a leg?

Anonymous
Miami, FL

A. I'd donate it and deduct its value from my income tax, but your question was if it could be used with a PC video card. No. Again, formats get in the way. If you have

Figure 7



Multiple Output Power Supply

a passive LCD screen, then it uses what's called dual scanning to paint the image on the screen.

With this method, the screen is divided in half — top and bottom — and scanned simultaneously to increase screen brightness. Active LCD displays usually don't dual scan, but use a fixed matrix of manipulated pixels that doesn't coincide with the raster pattern of a CRT scan. If, by a slim chance, you can find a video controller that supports an LCD screen and plugs into your motherboard's ISA slot, you'll still need to fabricate a cable to connect to your particular LCD panel.

PC Board Sound Connections

Q. I read your "Cassette to PC" answer in the September 2004 issue and found myself in the same boat; however, I don't need to listen to the source. I just need to find *one* line-out "stereo" plug that fits the earplug jack on my cassette boom box (good sound, as well as portable) and the line-in jack on my Sound Blaster sound card.

Unfortunately, the folks at RadioShack don't know anything about anything unrelated to selling cell phone service. The guys at Fry's are too busy to have time for such a trivial sale. So, my quest for the right connectors continues. Do you have the actual name of the plug combo I need and a source for them?

Bob
via Internet

A. The sound card uses a 1/8" (3.5 mm) plug and most boomboxes sport the same connector. This means the RadioShack 42-373 stereo patch cord should work. Portable CD/MP3 players uses a 3/32" (2.5 mm) connector; use a RadioShack 274-373 stereo adapter with the above cord for this connection. If, by some slim chance, your boombox has a 1/4" phone plug, you'll need a RadioShack 274-367 stereo adapter.

Multiple Power Outlets

Q. Did you, by chance, write about or remember seeing a power supply circuit that provides 5, 10, and 15 volts? It used an LM317 regulator to get 15 volts. From that, it used a resistance voltage divider to feed separate transistor emitter followers for the 5 and 10 volt sources. Any guidance will be appreciated.

Richard Ober
Baton Rouge, LA

A. No, but your description makes it very clear as to how the circuit looked. It also makes me think it's an older design — one that can be improved upon using today's cheaper, off-the-shelf voltage regulator chips.

My updated design in Figure 7 starts with a 7815 to provide the central 15 volt source. The 10 volt output is derived from a 7805 that is biased 5 volts above ground.

Let me explain. The 7805 chip references its output to ground — its ground. Now, if you float that ground above

the real ground, the output voltage of that regulator will be offset and held constant at the offset voltage plus 5 volts. The 5 volt output is self-explanatory.

Like the original design, the total output current is limited; this time, it's a collective 1 amp. That means you can draw 500 mA from the 5 volt source, 400 mA from the 15 volt source, and 100 mA from the 10 volt source — or any combination thereof. If you can live with 100 mA on the 10 and 5 volt outputs, the 7805 ICs can be replaced by the cheaper 78L05.

Precision 15-Minute Timer

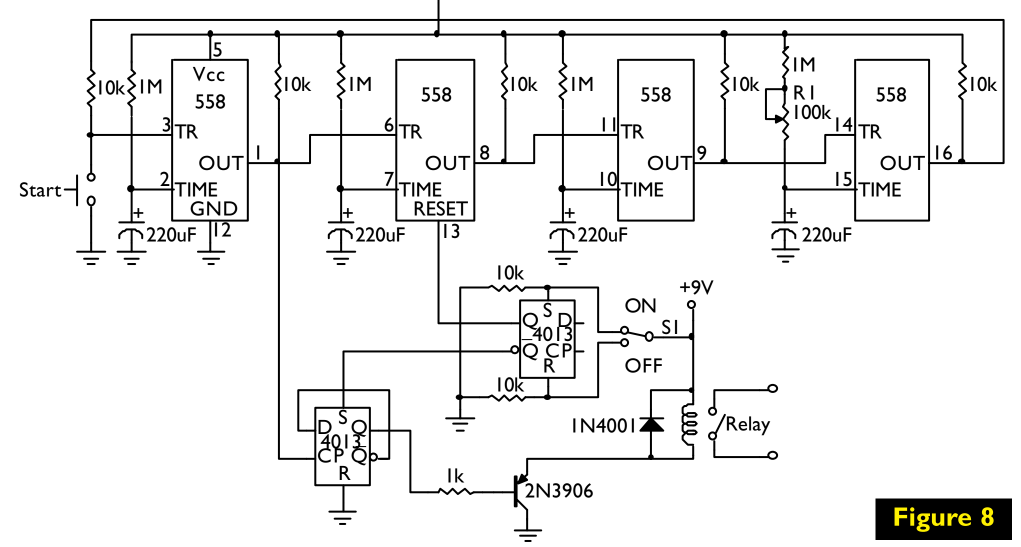


Figure 8

Optocouplers Demystified

Q. I'm fairly new to electronics and am self-taught through magazine articles like yours. In the "Outbuilding Timer" schematic in the September 2004 issue, you specify a MOC3020 optoisolator. Will any in the MOC30xx series work? What are the differences between, say, a MOC3031 and a MOC3020?

Dick B.
via Internet

A. The difference between the devices in the MOC30xx series is the current needed to light the LED and the breakdown voltage of the internal triac. The lower the LED I_F, the more sensitive the triggering of the triac. The series is also divided into random turn-on and zero-crossing turn-on. Table 1 defines each member in the series.

	Random			Zero-Cross			
LED I_F	250 V	400 V	600 V	250 V	400 V	600 V	800 V
30 mA	—	MOC3020	—	—	—	—	—
15 mA	MOC3010	MOC3021	MOC3051	MOC3031	MOC3041	MOC3061	MOC3081
10 mA	MOC3011	MOC3022	MOC3052	MOC3032	MOC3042	MOC3062	MOC3082
5 mA	MOC3012	MOC3023	—	MOC3033	MOC3043	MOC3063	MOC3083

Table 1. MOC30xx series optocoupler parameters.

MAILBAG

Dear TJ,

In the May 2004 issue, you designed a "precision" 15 minute on, 15 minute off timer with at least three ICs and, "a fistful of resistors and capacitors." How about one IC and just a few components?

Using a 4060 running in RC oscillator mode at 18.2 Hz and taking Q14 as the output, it will provide 15 minute on/off cycling and get rid of the 220 µF (eech!) capacitors. Using a custom crystal instead of the RC oscillator and adding a 4020 or 4040 can make

an even more precise timer.

Joe
via Internet

Response: *Joe, I have used this method in past columns. The 558 was used in this instance to mix things up and show the reader that there are alternatives to a digital world. — TJ*

Dear TJ,

Reading through the November 2003 issue, I ran across a request from Mr. James Tadlock concerning the 4N25 optoisolator. Having used this type of device in many of my previous designs, I have to say that you are correct in telling Mr. Tadlock not to ground pin 6 — otherwise known as the base. To ground the base would guarantee that the circuit would just sit there — dumb and happy, doing nothing.

However, the best way to handle the base in the 4N25 is to place a 100K resistor from pin 6 to ground. Adding this resistor will cause the I_{cbo} current to be bled off through the 100K resistor and will guarantee that the base is not pre-charged.

Norman A. Howard
Roseville, CA

Dear TJ,

In your May 2004 column, the Precision 15 Minute Timer (Figure 5 in the May issue) shows a 558 IC with the same pin numbers being utilized on all four timers. This can't be right, can it?

Phillip Milks
via Internet

Response: *Oops! This is what happens when you do a copy and paste of identical modules in a schematic and forget to follow-up with a proper pinout. Here's the corrected schematic* (Figure 8). *— TJ*

Dear TJ,

In the May 2004 issue, you give sources for downloadable data sheets. There is another one I like to use — **www.questlink.com** It is similar to ChipDocs, but free.

Bill
via Internet

Cool Websites!

Build a White-Box TiVo
For under $200.00, you can transform a PC into a full-featured personal video recorder.
www.techbuilder.org/article.htm?ArticleID=50808

Free clip art and document templates for Microsoft Word 97 on up. Templates include a simple will, power of attorney, birthday cards, Christmas newsletters, and more.
http://office.microsoft.com/clipart/default.aspx

http://office.microsoft.com/templates/default.aspx


Lowest Prices Guaranteed!
Greatly Expanded Line of
Quality Electronic Components
Celebrating
30
Years
Over 1,400 New Products Added
Semiconductors
Over 1,000 of the industry's most
popular, ranging from Analog to Zilog.
Power Products
Your source for high quality power,
whether you need medical grade or commercial grade.
NTE Replacement Semiconductors
The top 550 NTE quality products.
All in-stock at unbeatable prices
Intel CPU's & Motherboards
We've assembled the top proven performers in
microprocessors & motherboards to cover your requirements.
It's Worth Your Time
To Shop Jameco Again!
• We Guarantee the Lowest Prices
• Up to 64% Savings from List Price
• Over 99% of Products In Stock
• Product Offering Expanded 50%
These are just some of the latest additions!
Log on today for so much more
www.Jameco.com/NV8
Get your FREE catalog today!
Just mention VIP # NV8
1-800-831-4242
JAMECO®
ELECTRONICS
Brand Names & Generic Equivalents
Quality Components For Less
The Best Values
We search the world to
find you the best values!
The Lowest Prices
We guarantee the lowest
price in any catalog.
Greatly Expanded Products
We've added more than 50%
new products in less than a year.
Immediate Availability
Over 99% of products are
in stock for same day shipping.
Circle #42 on the Reader Service Card.

Electronics Showcase

Reader Feedback

(Continued from Page 6)

from the combination or optical mixing of red (700 nm) and green (520 nm) light in the right ratio at the right amplitude, but the perception of yellow can just as well result from a true yellow (580 nm) light stimulus. I suspect that the yellow phosphors in Shuji Nakamura's white light experiment were radiating in the 580 nm region and not in the red and green.

The perception of white can be achieved by mixing any one of many sets of just two colors, as well as by mixing the three additive primaries. Blue and yellow light compose one of those sets. This can be easily verified by checking the CIE chromaticity diagram.

Tom Gordon
via Internet

Dear Nuts & Volts:

I've been a *Nuts & Volts* reader for over a year now. I really like your magazine and I would like to make a suggestion for an article — batteries, including NiCads, NiMH, alkalines, Li-Ion, etc.

It would be nice to know the what, where,why, when, and how for using each type. How do they compare? How do you charge them? What exactly is trickle charge? If I want to make a rechargeable gizmo with a base, what circuitry should I use? How do I effectively test them?

For example, a completely dead NiCad at rest on a shelf for several weeks will get a self-boost that will — if we use a cheap tester — indicate that it is somewhat good and not completely dead. However, if you hook that battery up to a small bulb, it will light up relatively brightly for about 5 seconds, then go completely dead. Weird.

Also, I understand that battery life is measured in mAh, but why do digital cameras use up regular batteries so fast (I mean, really fast!), but not so with Li-Ion or Ni-MH? Doesn't 5 V = 5 V from battery to battery? Can you recharge regular batteries? A few years ago, a TV ad was selling the BuddyL charger. Did that really work? If so, how?

I hope these questions inspire someone to write a super, educational article on the wide world of batteries.

Martin Beaudry
Montreal, Canada

Dear Nuts & Volts:

In general, your magazine serves the needs of the professionals, the experienced hobbyists, and the novices in electronics. Minor errors are sometimes noted and almost always corrected by readers and/or your own editors.

However, your September 2004 issue contains one article that goes far beyond just a few typographical errors or oversights. Unfortunately, it is an article in the "Just For Starters" section, aimed at beginners. For example:

- The units of bulk resistivity are ohm-cm, not ohms per cm.
- A junction is not P-type or N-type; a region is.
- An SCR is *not* a silicon<u>e</u> controlled rectifier!

I hope that, in the future, more thorough reviews of such articles are done — especially those that are intended to help a beginner understand something about electronics.

Jim Galvan
Corvallis, OR

ERRATA

*Due to an oversight, the parts list was accidentally left off of Ron Newton's "Electronic Sniffer" project in the September 2004 issue. It is posted on the Nuts & Volts website at **www.nutsvolts.com** for download along with the source code and PCB layout files.*

by Walter Krawec

Palm Programming: An Introduction

Learning to Use PocketC

This Month's Projects

The Fuzzball Rating System

To find out the level of difficulty for each of these projects, turn to Fuzzball for the answers.

The scale is from 1-4, with four Fuzzballs being the more difficult or advanced projects. Just look for the Fuzzballs in the opening header.

You'll also find information included in each article on any special tools or skills you'll need to complete the project.

Let the soldering begin!

Developing applications for a Palm device can be costly. Not only that, but it can be confusing and difficult. However, it doesn't have to be all that (well, not costly, anyway). Using free or very low-cost tools, you can get underway almost immediately and be developing programs that run on the Palm in no time.

What's out there, other than Code Warrior, which costs $399.00? Don't get me wrong, Code Warrior is an excellent compiler ... it's just the price that's the problem. A few years ago, after buying my first Palm device, I came upon that very question. Palm developed a superb SDK or Software Development Kit, which is even available as a free download. (See the Links sidebar for the URL.) That's nice and all, but you still need a compiler to compile all the files that came with the SDK, not to mention your own program files.

So, after a few more weeks of using my Palm as a simple PDA, I stumbled upon the answer! I found out about PocketC by OrbWorks. (Again, see the sidebar for the URL.)

"What is PocketC?" you ask. Well, basically, it's a C compiler that, instead of running on your PC, runs on your Palm.

As with everything, PocketC has its advantages and disadvantages. First, let me list the advantages. The first of which — and what most people will think of as the best advantage — is its price. It only costs $18.50! That's a very good price for a compiler with as many features as PocketC.

The next advantage is its ease of use. To use PocketC, you simply install it on your Palm device and you are ready to compile programs instantly. There are no libraries to set up and no compiler options to play around with to get it working right. PocketC simply runs "right out of the box."

So, those are the advantages — some of them, anyway. Now for the disadvantages. PocketC is not a true compiler. What I mean by that is that it won't compile your C program into a real Palm program that any Palm device can easily run. Instead, what PocketC does is compile your C program into a "byte code" that you then need PocketC to actually run. Because of this, PocketC programs are slower than real Palm programs.

I once wrote a simple terminal editor program for the Palm using PocketC and that program had a lot of trouble keeping up with serial input at only 9,600 bps. However, as long as you are careful to optimize your program in every way possible — which is something you should do anyway — you shouldn't run into too many problems.

Another of PocketC's disadvantages is that you need PocketC installed on your Palm in order to run programs that were compiled with it. Again, this is because PocketC compiles your program into a "byte code" and not a real Palm executable. However, this isn't too bad because Orbworks (the creator of PocketC) develops two versions of PocketC — the full version and the free version.

The full version will cost you $18.50 and actually allows you to compile programs. As the developer, you would have to buy this

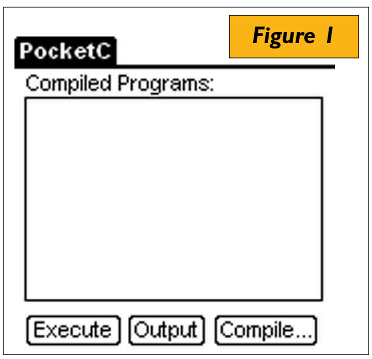


Figure 1

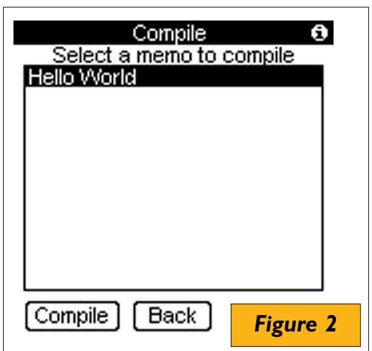


Figure 2

version to make anything with PocketC. The free version, on the other hand, is only able to run previously compiled programs. This version cannot compile new programs. If, for example, you developed an application that you wanted to distribute to other people, you could simply package your “byte code” file along with the free version of PocketC. That way, the user simply downloads and installs both files at the same time, without any headache.

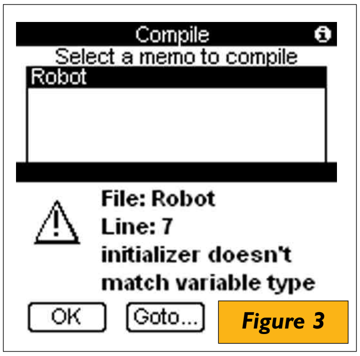


Figure 3

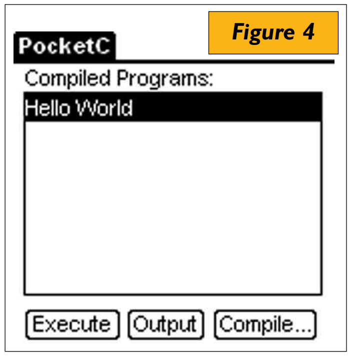


Figure 4

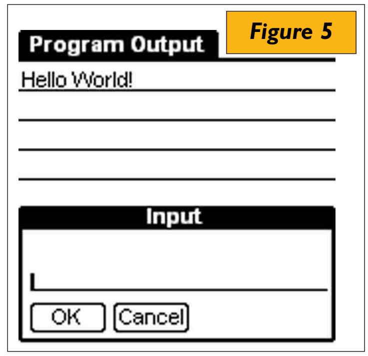


Figure 5

Hello, World!

Okay, I’m done talking about how great PocketC is. Now, let me prove to you how easy it is to use PocketC. To do this, I’m going to show you a very basic program — Hello, World. (What, you were expecting something else?)

If you look at Listing 1, you can see the “Hello, World!” program in its entirety. Yes, that is all you need to create a PocketC program; there are no other files needed. You don’t need to go through any documents to figure out how to get the compiler to work. Now, I’m not going to bore everyone by going through the listing line by line; however, there are a few things I must point out.

First off — and this could possibly be the most important element in PocketC — is that beginning remark (the line starting with the //). Now, most of you probably thought that this was just to tell everyone looking at the code that this was a “Hello World!” program — not so. You see, PocketC needs a way of determining whether a text file is a PocketC program or just a regular note file. The way PocketC does this is by looking at every memo file on your Palm and seeing if the first line is a remark. If it is, then that memo is considered a PocketC program; otherwise, it’s ignored.

Now, after the //, you simply put down the name of the program. Our “Hello World!” program is called “Hello World1” If, for some reason, you wanted to call it “abc,” then the first line of the program should be “// **abc**” instead of “// **Hello World**!”

Well, I guess that was actually only one thing to point out. The rest is pretty straightforward; you have your **main()** procedure which is ... No, I’m not going to go through it line-by-line. The PocketC documentation does a much better job at explaining things than I ever can.

Now that you have your PocketC program typed up, you have to actually compile it. However, doing that is even simpler than typing up the program. Simply load up PocketC and you will be presented with a screen, as shown in Figure 1.

You will notice that you have three buttons to choose from — **Execute**, **Output**, and **Compile**. I’m pretty sure you can guess what **Execute** and **Compile** do and you don’t have to worry about the **Output** button.

So, we simply tap the **Compile** button and PocketC will then give us a list of all the PocketC programs currently on your Palm. Again, PocketC determines whether or not a file is a PocketC program by that first remark. As shown in Figure 2, PocketC found one program, our “Hello World!” program.

So far, so good! To actually compile the program, we select it from the list and tap **Compile** again. Now, one of two things will happen. If there are any errors in your program, something like Figure 3 will pop up. Basically, PocketC will give you the error type, where the error is located and — best of all — a **Goto** button. Tapping that button will take you directly to the error line so that you can fix it right then and there and be back in PocketC in no time.

However, if the program did compile successfully, you will be taken back to PocketC’s main menu. This time, though, the “Hello, World!” program will be listed in the program list box (as shown in Figure 4). That’s it! You now have a working PocketC program. To run it, simply select it from the list and tap the **Execute** button.

That was fun. We built a simple — yet functional — Palm OS program. Now what do we do? Well, in order to create more complicated and infinitely more interesting programs, we’ll need to know what else PocketC is capable of. To explain as many features as possible, I’m splitting

Listing 1

```
// Hello World

main()
{
    puts("Hello World!");    // prints "Hello World" to the screen
    gets("");                // waits for the user to tap OK before exiting
                             // that way the user has time to read the message
}
```

this article into several sections, such as “Basic I/O” and “Serial I/O.” In these sections, I’ll talk about some of the various procedures and other elements that are available to you when using PocketC.

Basic I/O

Every program, no matter how simple, needs some basic I/O commands. In our “Hello, World!” program, we used the **puts()** command, but this isn’t the only output command available to you in PocketC. In fact, PocketC provides you with a plethora of commands that deal specifically with input and output. For example, simply printing text to the screen using the **puts()** command might not be enough. At times, you’ll probably want the message to “pop out” at the user (such as an error

Listing 2

```
// Remote

main()
{
    int size;
    int size2;
    int i;
    seropen(9600, "8N1C", 100);     // opens the serial port so we can use it
    graph_on();                     // turn the graphics on in PocketC
    title("Remote");                // now the text "Remote" will be printed at the top of the screen
    clearg();                       // clear the screen

    size = getscreenattrib(1);      // get the size of the screen (in pixels)
    size2 = size / 2;               // and divide it by two. Then store this "Midpoint" in "size2"

    line(1,0,size2,size,size2);     // split the screen into four sections
    line(1,size2,0,size2,size);     // by drawing two lines

    text(10,40,"Back");             // and write some text to show what each section is
    text(size-70,40,"Forward");
    text(10,size-40,"Left");
    text(size-70,size-40,"Right");

    while(1)                        // an infinite loop
    {
        i = event(1);               // wait for something to happen
        if(i == 2) {                // a pen down command?
            i = peny();             // yes, so use peny() and penx() to determine where the pen is
            if(i > size2) {
                i = penx();
                if(i > size2) {     // lower right
                    sersend("4");   // RIGHT command
                }
                else {              // lower left
                    sersend("3");   // LEFT command
                }
            }
            else {
                i = penx();
                if(i > size2) {     // upper right
                    sersend("1");   // FORWARD command
                }
                else {              // upper left
                    sersend("2");   // BACK command
                }
            }
        }
        else if(i == 5)             // page up key
            sersend("0");           // STOP command
        else if(i == 6)             // page down key
            break;                  // quit
    }
    graph_off();                    // don't forget to close the graphics...
}
```

message). For this, you can use the **alert()** command, which will actually create a separate dialog box and output the text there. Even better is the **confirm()** command, which not only creates a separate pop up box, but also adds "Yes" and "No" buttons. Using this command, you can easily receive user input without any hassle.

This should be enough "basic" output commands for most programs, but what about input? While not all programs require any input, most do, and PocketC equips the programmer with several of these much needed commands. First and foremost is the **gets()** command. When this command is called, the Palm will create a new input box with two buttons: "OK" and "Cancel." Here, the user of your program will be able to type in an entire string that the **gets()** command will return if the user taps "OK." If the user taps "Cancel," then the procedure returns an empty string. Not only does this procedure create a simple input box, but you can also give the procedure a string of text to print out on the screen.

For example, **gets("Type in something")**, when run will create a pop up box asking the user for input. However, the text inside of that box will read, "Type in something."

There are also several **gets()** variants, such as the **getsd()** procedure. This new procedure does the same thing as **gets()**, except that you can give the input box a default value. Also available to you are the **getsi()** and **getsm()** commands, which act like the original command, except that you can tell PocketC exactly where to place the input box (instead of the default bottom of the screen). The **getsm()** procedure also adds the ability to control the size of the input box in addition to the other features of the **getsi()** procedure.

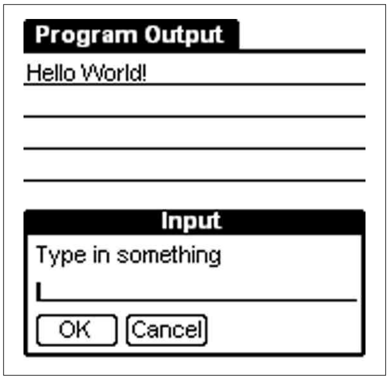


Figure 6. A screenshot showing the gets() procedure.

Figure 7. A screenshot showing the resulting memo file from the "Creating Memo Files" section.

Events

We now know how to output text to the screen and receive user input, but how do we know what the user is doing to the Palm at any given time? For example, a program might need to know when a user taps the screen or presses the "Calendar" button. This is all handled through the Event System in PocketC.

While there are several commands that can be considered a part of the event system, the command you'll probably be using most is the **event()** command. This command will basically check to see what the user is doing and report back with the number code of the event. Say, for example, the user taps the screen. The **event()** procedure will return the number **2**. Later on, however, the user presses the Page-Up key. In this case, the **event()** procedure will return a **5**. All in all, there are 18 different events that this procedure can track. If you need to know what they all are — and I'm sure you will — take a look at the PocketC documentation.

Figure 8. My Palm device with cradle.

Figure 9. All you need to easily develop Palm applications. Neither the computer nor the Palm has to be very powerful.

In addition to the **event()** procedure, the Event System contains several other useful functions, including the **penx()** and **peny()** procedures. These are two procedures you'll probably be calling a lot if you create a Palm GUI. To put it simply, these procedures return the x and y position of the pen (or stylus).

Let's say that your program calls the **event()** procedure, which, in turn, returns the value **2** (pen down event). Now, all we know is that the pen is actually touching the screen. Maybe this is enough for some cases, but — most of the time — you're going to want to know *where* the pen is touching the screen. To do that, simply call the **penx()** and **peny()** procedures and they'll tell you everything you need to know.

Now, you're probably aware of the fact that, when you press the "Calendar" button or any other short cut button, the appropriate application loads up automatically. This is because Palm OS is constantly checking these buttons ... even when your program is running. So, ordinarily, even though you're calling the **event()** procedure to check for the status of these buttons, Palm OS will take over the second any of them are pressed.

That's where the **hookhard()** procedure comes in. If your program needs to use these predefined buttons, you'll have to call this procedure before anything else is done. If you pass a **1** to this procedure, all of these buttons will be processed through the **event()** procedure *before* going through the OS. Otherwise, as I mentioned before, as soon as any of these keys are pressed, the OS takes over and launches another application.

Strings

Easy string manipulation is often a weak spot in C compilers, but this is not the case in PocketC! In fact, working with strings is as easy as working with any other variable type. Just define them and you can set them up as easily as an integer. Of course, this isn't nearly enough for more advanced programs, so PocketC packs a bunch of easy-to-use string procedures.

Included with these procedures is the **strlen()** procedure, which returns the length of the string. Also, the **strupr()** and **strlwr()** procedures are available to you. These procedures will return any string you pass to them in either uppercase (if you use **struper()**) or lowercase (if you use **strlwr()**). On top of all that, you can also call the **strstr()** procedure, which will actually search through a string for a substring. If the substring is found, the procedure will return the starting character position of that string.

Mathematics

Of course, a compiler is only as good as its math capabilities and PocketC packs quite a punch in this area. However, while PocketC has several math functions, this functionality doesn't come naturally to the compiler. Instead, all of the math procedures are defined in an external library that you will need to install. Remember when I told you that there are no external libraries to install? I lied.

Well, actually, not really. I meant that you didn't *need* any external libraries to get started. However, if you want to add functionality to the compiler (and I'm sure you will), you will, of course, need some external definitions for things.

Fear not, though, for MathLib is here! In fact, MathLib actually comes with PocketC, meaning that, when you

Links

Development Sites

www.orbworks.com
Orbworks is the developer of PocketC. On their site, you can find information on PocketC, as well as other tools related to it. Plus, they have a great support forum. So, if you have a question, you can always ask.

www.palmsource.com/developers
Palm's developer site. Everyone interested in developing for the Palm should visit this site.

www.geocities.com/retro_01775/PToolboxLib.htm
Even though I didn't talk about PToolBoxLib in this article, I have to mention it. PToolBoxLib is a free graphics library for PocketC. Basically, if you want to do anything involving graphics in PocketC, you need PToolBoxLib.

http://www.geocities.com/waltsrobots/page3.html
My robotics website. If you have a question, don't hesitate to Email me.

download the PocketC files from OrbWorks, you're also getting this library full of mathematical knowledge. To use it, simply install it as you did PocketC. You should keep in mind, however, that if your program does use MathLib and you do plan on releasing your finished program, not only will you have to tell the user to install the free version of PocketC, you'll also have to tell them to install MathLib, but that's not too bad because MathLib comes bundled with the free version of PocketC.

So, now that we have that out of the way, let's get to the math! Once you've installed MathLib, your Palm program will be able to access a multitude of mathematics functions, including the trigonometric functions such as **sin()**, **cos()**, **atan()**, and all the others. Also included in MathLib is the **pow()** function, which takes two values (x and y) and returns x^y. That's not all — you also have the **sqrt()** function (which takes the square root of the argument), a couple of log functions, and even some random number generator procedures.

Sound and Time

Every Palm device has a buzzer of some sort. (At least, I have never seen or heard of a Palm that couldn't beep, somehow.) So, in order to use these buzzers/beepers/speakers, PocketC contains several easy-to-use procedures that you can call.

First up is the **beep()** command. Short, sweet, and to the point, this procedure simply generates a tone of some sort. However, it doesn't just make a single note on your device's buzzer. Oh no, you have up to seven different sound effects to choose from. By passing a **1** to this command, your Palm will give you an "info" sound or, if you pass it a **5**, you'll get an "alarm" sound. If you would like to learn more about all these fascinating sound effects, please refer to the PocketC user manual under the appropriately labeled "Sound" section.

Along with the **beep()** command, you also have a **tone()** command, which will generate a sound using the specified frequency and duration. While slightly more complicated, this gives you much more control over the generated sound when compared to the **beep()** command. However, if you use the **tonea()** function, not only do you have control over the frequency and duration of the sound, but you are also able to specify the volume the tone will be played at. Pretty cool, 'eh?

While there may be a nice variety of sound commands, there aren't many time commands. However, what more do you really need, other than a command that will report the number of seconds that have passed using the **seconds()** command or the number of clock ticks since the last reset using the **ticks()** function.

Creating Memo Pad Files

Memo files are Palm's equivalent of the text file. As such, they can be used for many purposes, including debug logs and even as a place for your application to save some of its options. Of course, to take advantage of these simple files, PocketC contains several easy-to-use procedures that you can call.

First off, before you can write to a memo file, you must create it. This can be done with a simple call to the **mmnew()** function. This will create a new and empty memo file. Why don't we give this procedure the file name that we want? Palms are very simple when it comes to file management. Instead of each memo file having its own unique name, they are simply named after the first line of text they contain. So, a memo file that contains the text:

Hello
World!

will simply be named "Hello."

We just created a new memo file, but have yet to write anything to it. In order to write something to this new file, we call the **mmputs()** procedure. This procedure works just like the **puts()** command we used earlier, except that, instead of printing the text to the screen, the text will be written to our new memo file. When you are done working

with this memo file, don't forget to call the **mmclose()** procedure, which will close the file.

What if we want to read the text from a memo file that's already been created? Well, first you call the **mmfind()** function, giving it the file name of the memo; remember, the file name is the first line of the file. This command will look through every memo and, if it finds the correct one, that file is opened. Otherwise, a **0** is returned, meaning that some error occurred.

After opening the file, you can use the **mmgetl()** procedure, which will return the entire line of the memo file as a string. If you ever want to start at the top of the memo file again (as if you just opened it), simply call the **mmrewind()** procedure. Also, if you don't like the memo file you're working with, you can simply call the **mmdelete()** function, which will close and delete the file.

Serial I/O

That's really all you need to know to create most Palm applications, but what if you want your Palm to communicate with some other device, such as a robot or another computer? While there are several ways to solve this problem, the simplest is to use your Palm's onboard serial port.

Before you can send or receive any data over the serial port, you'll need to open it using the **seropen()** command. This command takes three arguments: First, there is the baud rate, which is an integer (such as 9600). The next argument is a string of flags including bits, parity, stop bits, and flow control. Usually, you'll set this to "8N1C." The last parameter is a timeout integer, which tells PocketC how long to wait (in 1/100 second) between bytes for data.

Now that the serial port is opened, you'll be able to send and receive data. Use the **sersend()** command to send a single byte of data. For example, **sersend("A")** will send the character "A" over the serial port. To receive data, call the **serrecv()** command, which will return the received data (if any) as an integer (one byte's worth). Don't forget that, when you open the serial port, you have to close it. To do this, simply call the **serclose()** procedure before exiting your program.

Conclusion

I sincerely hope you enjoyed this article. I also hope it encouraged you to start programming a Palm. What I talked about here was just a small amount of what's possible with the Palm. So, good luck with all your projects — Palm related or not — and have fun! **NV**

Circle #52 on the Reader Service Card.

by Ron Newton

The Digital Energy Saver

Marking Time to Save You Money

Even though America is an affluent nation, this does not mean we need to squander our money. Consider that the price of gas approached $3.00 in some areas this summer and that energy rates will continue to climb this winter. The cost of the energy used must be counted, but the wear and tear on equipment is just as important. Mechanical timers for shutting down equipment have been around for many years, but are often short lived or not accurate and cost about $30.00. They also have to be purchased in the range needed, e.g., five minutes, one hour, and 12 hours.

One of the most common problems I encounter is leaving my soldering iron on, only to discover days later that I have ruined a good tip and wasted energy. I have a mechanical timer in my machine shop for a small air compressor.

Recently, the timer snapped because the timing position was on to the "on all the time" position. It must have run for two weeks, as it was on when I returned from vacation. Closet lights are also a problem, particularly with children.

This energy saver project is inexpensive (less than $15.00) and provides a solution for all of the above, in addition to coffee pots, irons, and one hundred and one more applications. It was designed for mounting in the wall switch, replacing the standard on and off switch. This makes it ideal for hallways and closets (see Figure 1). It is capable of handling 3 amps, but can be adapted to up to 10 amps. This project was designed be a universal timer with a range from seconds to days, depending on the application. The timing is variable from 1 second to 63 days. For soldering irons and coffeepots, you can mount it in a project box using the ends of a small three pronged extension cord.

Figure 1. The completed unit in the Hammond enclosure.

Methodology

The heart of the timer uses a Microchip PIC16F627A. The power supply is simple using a 1 µF, 250 volt capacitor (C1) and two diodes (D1-D2). AC passes through the capacitor; the two diodes rectify the AC to DC and a 220 µF capacitor stores the power. A zener diode (Z1) controls the DC voltage to 5.6 volts. The timing is provided by the 60 Hz signal from the 110 volt line. This provides an accuracy of 1/60 of a second.

An eight-position dipswitch provides the hardware for programming the timing. Switches one to six provide counts from 0 to 63 or, more practically, 1 to 60. Switches seven and eight provide for programming in seconds, minutes, hours, or days.

The AC turn-on device is a solid-state triac device developed by Sharp. It turns on by applying 1.2 volts to its internal LED. I have used this triac on small motors and fluorescent lights and it works well with resistive and inductive devices. Sharp also makes similar devices with snubbers for inductive loads and has zero crossing models.

The device described turns on by the push of a switch. The unit will time to the predetermined programming and then turn off. Pushing the switch again will turn off the power when in the timing mode. Want to have the light stay on longer than programmed? Just hold down the button until the light comes on and the light will remain on until the button is pushed again. This disengages the timing circuit. The unit was designed to be fail-safe, in case of a power disruption; it defaults to the off position in such a case.

Construction

The PIC microprocessor will need to be programmed using a programmer. The source program and the object

files are on the *Nuts & Volts* website at **www.nutvolts.com** If you don't have a programmer, the chip can be purchased along with a commercial board. Although the board is simple enough to be wire wrapped, it is better to use a printed circuit board.

There are two jumpers required if you etch a single-sided board. Solder these first. Pay attention to the polarity placement of the three diodes and the 220 µF cap. Both the diodes and the resistors are placed standing up for space consideration. R3 is a resistor network. Make sure the common pin of the resistor net goes in the location marked "pin-one." The printed board layout has extra holes, as capacitors come in different sizes. Also there is an extra hole for the resistor net, allowing you to use a 10-pin version.

When mounting the triac, hold the triac with needle nosed pliers at the thick part of the lead and bend the remainder of the leads down at right angles using your fingers. Mount the triac with a 4-40 screw and nut. Cut 6" of red, white, and black #18 stranded wire and solder to the board to their respective locations. I recommend using an 18-pin socket for the microprocessor if you are going to change the programming of the chip.

The switches come in different lengths. Using a short one adds security against bumping and children, but requires an instrument such as a pen or pencil to turn it on. The 7 mm switch is ideal for a light switch plate. The switch is soldered to the opposite side of the component side of the board.

Before inserting the microprocessor, secure the board from moving and connect 110 volts to the white and black wire. The red wire is the switch's 110 volts. Measure the voltage between pins 5 and 13 of the microcontroller. This should be a 5.6 volts.

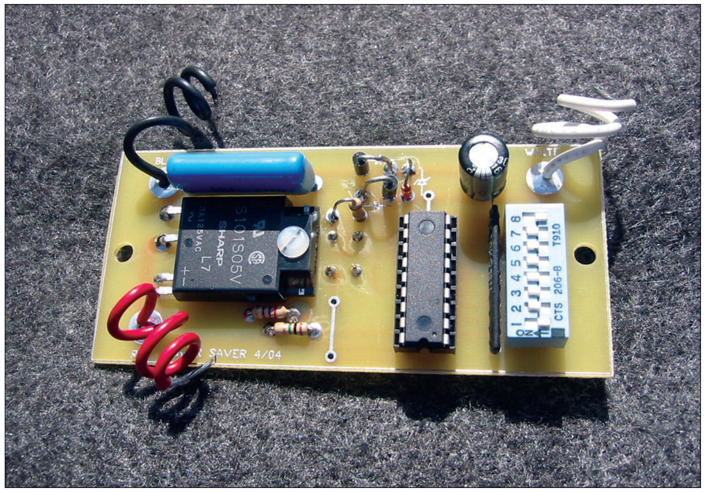


Figure 2. The parts on the PC board.

Programming

First, determine the number of seconds, minutes, hours, or days you wish the appliance to be on. The dipswitch is a binary switch, which counts in the base of two. The first six switches program up to 63 units. The last two sets of the switch provide counting in seconds, minutes, hours, or days. It is easier to use the look up table included on the website. For example, if you are programming for a closet and you want the unit to time for five minutes, set the switches as follows: 1 off, 2 on, 3 off, 4 on, 5 on, 6 on, 7 off, and 8 on.

Switch Plate Installation

The switch plate is made out of blank electrical box cover. Do not use metal plates. When purchasing the plate,

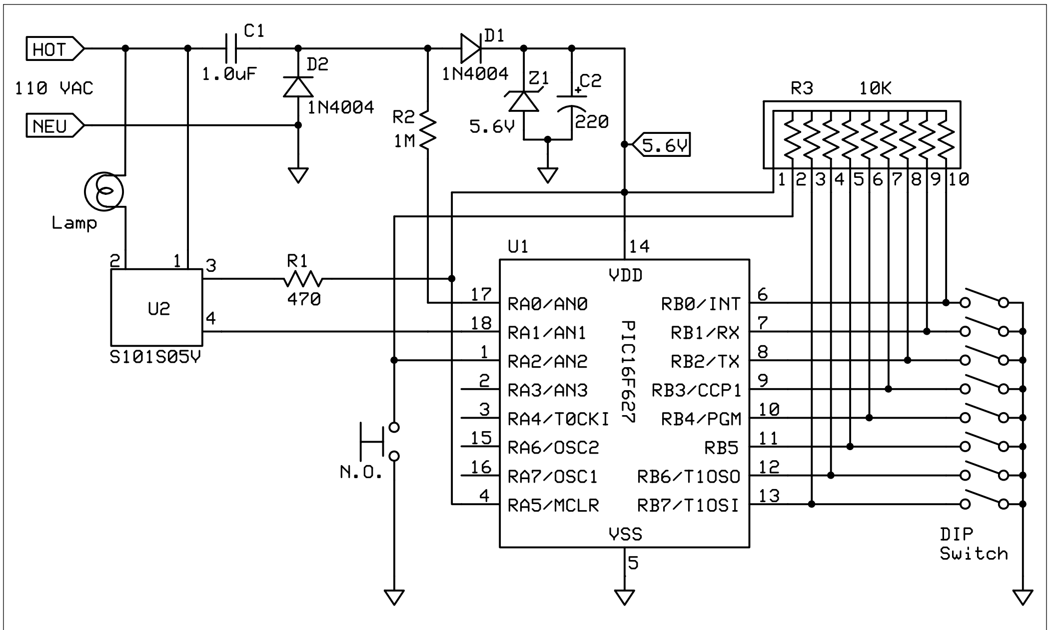


Figure 3. The schematic with the PIC controller, DIP switch, and switching triac.

they will cause warping of the circuit board. Nylon plates will drill better. Place a piece of masking tape over the center of the cover on both sides. This will help prevent chipping. Locate the center of the plate by using diagonal lines. Drill using a 3/16" drill.

Turn off the circuit breaker. Wire the unit in using wire nuts. The white wire goes to white and the black to black — the hot side of the switch. The red wire goes to the other black wire that goes to the appliance. Place the switch plate on the unit with the switch protruding through the hole. Using look for the type which has built-in spacers on the mounting holes and not the bar spacers in between, as the two screws that came with the switch plate, secure to the electrical box.

Parts List

Semiconductors

IC1	PIC16F627A microcontroller
IC2	S101S05V 3 amp triac
D1, D2	1N4004
Z1	1N5232 Zener, 5.6 V 500 mW

Resistors (all are 1/4 watt, 5%)

R1	470Ω
R2	1M
R3	10K network, bussed, 10-pin (like Bourns 4610X-101-103)

Capacitors

C1	1.0 µF 250 WVDC metallized
C2	220 µF 16 WVDC radial-lead electrolytic

Additional Parts and Materials

Momentary tactile switch, 7 mm (S1), Eight position DIP switch (S2), Plastic enclosure, Hammond 1591B series, DPDT relay for higher current switching (RadioShack 275-217).

The following items are available from Ron Newton, 2230 Damon Rd., Carson City, NV 89701, (775) 885-8842, Email: sjnewt@att.com — pre-programmed IC1, $10.00; etched and drilled PC board, $15.00. Please add $5.00 for shipping and handling within US and Canada. NV residents must add appropriate sales tax. Pay Pal is welcome.

Utility Box Mounting

The unit will mount nicely in a Hammond 1591B series enclosure. Use a pre-drilled switch plate for a template. Cut a small, three-pronged extension cord 6" from the female end. The male end's length is at your discretion. Mount the cords, placing two strain relief bushings into the ends of the box.

Wire the white wire to the wires that go to the wide blade of the plug and the wide blade hole of the female side. Wire the black to the small blade of the male plug. Wire the red wire to the small blade hole of the female side. Tie the ground plug wires together. You will need to add two .125" spacers between the mounting screws and the board.

Need More Power?

If you need more than three amps, you can use a higher rated Sharp triac, but it will have to be heat-sinked. The easier way is to use a RadioShack 275-217 double pole, double throw, 10 amp relay and mount the unit in a large box. Connect the output of the triac to the relay connections. Make sure that you increase the gauge of wire to prevent overheating. **NV**

Multiplexing to Get More Outputs

An Introduction to Output Pin Decoding and Expansion

Assuming that your system has some digital outputs, but you need more outputs for the current project you're working on, what do you do? This article will explain how to add as many digital outputs to your system as you desire. The types of systems I'm referring to are typically microcontrollers, such as the 68HC11, 68HC12, 68332, PIC chip, BASIC Stamp, ATMEL processor, and a host of others. This technique can also be applied to the parallel port output of a PC. I will present this information in a generic manner, which should apply to any target system. The usage examples will be in pseudo-code and then in assembly code, which are specific to the 68HC11, but this could be easily adapted to any other system.

The Concept

You can use a limited number of digital outputs — along with a simple digital circuit and program — to control as many digital outputs as you desire. The type of circuit we're talking about here is a multiplexer — MUX, for short. With a MUX circuit, you take your digital outputs from your system and group them into two types: data lines and address lines. Depending on the number of system digital outputs you have to work with and how you group them into address lines and data lines, you can get various numbers of resulting MUX outputs, plus varied complexity of the required digital circuit.

Let's walk through an example using eight system digital output lines to see the possibilities:

The top row in Table 1 shows us that, if we use all eight lines as data lines, we get eight resulting output lines (and our circuit is very simple — just wires). The next row uses one of the lines as an address and the other seven as data. With the address set to 0, we can specify seven data values and then, with the address set to 1, we can specify seven more data vales, giving us 14 resulting output lines. Continuing down the rows, we see that the same eight system outputs can generate up to 128 resulting outputs. Wow! There is a generic equation to determine the resulting number of outputs we would get. Given **N** as the total number of system outputs, **A** is the number of address lines, **D** is the number of data lines, where **A + D = N**, we can calculate:

$$\mathbf{2^A * D = R} \text{ (the resulting output lines)}$$

Choosing N = 8, we can vary A, which generates D and R.

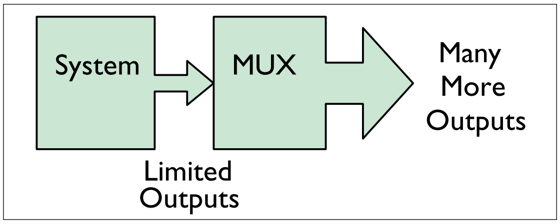
We can clearly see in the plot in Figure 2 that, the more lines we use as addresses, the more resulting outputs we will generate. We can, using the above equation, see the number of outputs we can get for different numbers of address and data lines.

However, there is a price to be paid for this; the complexity of the required digital circuit and the complexity and time required to execute the resulting driving program go up with the number of address lines. Another factor is the type and configuration of the digital chips that are available to use in the support circuit. In a minute, we will look at practical limitations and implementations.

Consider using from four to 12 output lines. Figure 3 shows how many outputs could be generated. The maximum for 11 bits is 1,024 and — for 12 bits — it is 2,048, which is a very large number of outputs from a small number of original output lines

Before we jump into building a 12-to-2,028 output circuit, let's first consider the building blocks available and the hardware/software complexity required, so we can design a

Figure 1. Concept of system outputs generating more outputs using a multiplexer circuit.



Starting Outputs	Address Lines (A)	Data Lines (B)	Resulting Outputs
8	0	8	**8**
8	1	7	**14**
8	2	6	**24**
8	3	5	**40**
8	4	4	**64**
8	5	3	**96**
8	6	2	**128**
8	7	1	**128**

Table 1. Address/data combinations for eight-bit output.

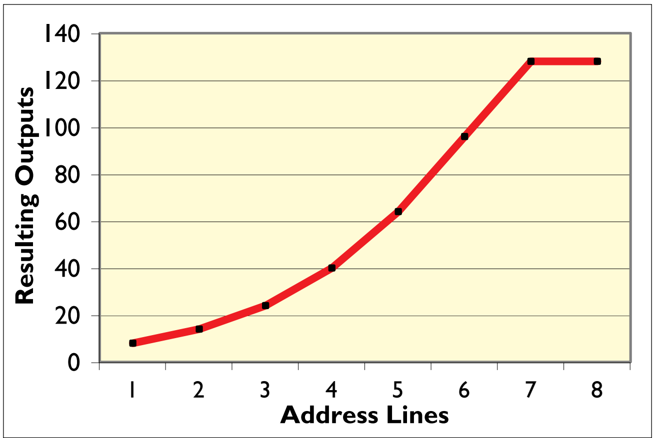


Figure 2. Plot of data from Table 1. Address to resulting outputs for an eight-bit multiplexer.

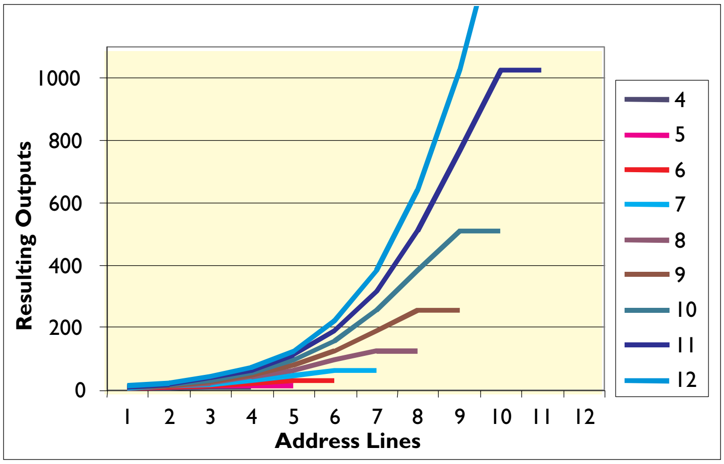


Figure 3. Possible number of outputs for four through 12 lines.

reasonable system that can be easily built and used, but will also give us a good expanded output capability.

Circuit Building Blocks

The digital chips available that we will use to build up our multiplexing output circuit are the 74HCnnn family of chips, but similar functionality can be found in the other 74-series (74LS, for example) and also in the old 4000-series CMOS devices.

Address Lines

For the address lines, we need to "fan out" the address into N control lines. We'll see why in a moment. The type of device we want here is called a "decoder," also referred to as a "demultiplexer." These take **A** address lines and produce $\mathbf{2^A}$ outputs, where one output as specified by the address is at one logic level (typically Low) and all the other outputs are at the other logic level (typically High). The two most-used 74HCnnn devices for this are the **74HC138** (1-of-8 decoder) and the **74HC154** (1-of-16 decoder). The '138 (short for 74HC138) takes three address lines and produces eight control lines, while the '154 takes four address lines and produces 16 control lines. The typical logic symbol for such a device is shown in Figure 4.

The three address lines are A0, A1, and A2. There are also three enable lines that can be used to configure multiple '138 devices to directly handle up to six input addresses. To enable a device, tie E1 and E2 low and E3 high. The eight generated outputs are Y0 through Y7. The circles on the diagram indicate that the active level for that pin is inverted, thus we can see that E1 and E2 should be low (inverted), E3 should be high, and the output addressed in Y0 to Y7 will be low (all other seven outputs will be high).

Table 2 is a truth-table for the 74HC138, showing the states of the outputs for all input combinations. The '-' entries are "don't care" states. **H** is logic high or 5 volts. L is logic low or 0 volts. Note that any enable input can disable the device.

Data Lines

For the data lines, we need to capture the data and hold their values until we want them to change. The type of device we want here is called a "latch" or a "flip-flop" (FF). These devices take **D** data lines and will capture their values when commanded to do so. The commonly used 74HCnnn devices for this are the **74HC74** (dual flip-flop, two in one chip), **74HC173** (quad flip-flop, four in one chip), **74HC174** (hex flip-flop, six in one chip), and the **74HC374** (octal flip-flop, eight in one chip).

These devices have data inputs, data outputs, a clock input, and output enable inputs. The data on the inputs is captured and presented at the outputs when the clock input goes from a low to a high state. Tie the output enable(s) to enable the device (low on the '374). Figure 5 shows the typical logic-symbol for a 74HC374 eight-bit latch.

There is a device — the **74HC574** — that is identical in

Figure 4. Decoder logic symbol.

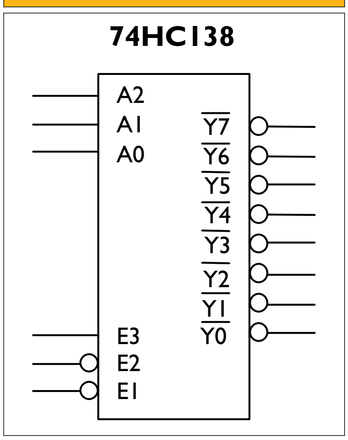
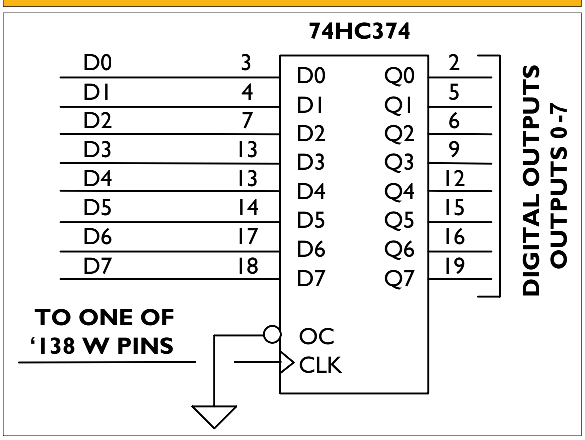


Figure 5. Eight-bit latch logic symbol.



Address Lines			Enable Lines			Output Lines							
A2	A1	A0	E3	E2	E1	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
-	-	-	-	-	H	H	H	H	H	H	H	H	H
-	-	-	-	H	-	H	H	H	H	H	H	H	H
-	-	-	L	-	-	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	H	H	H	**L**
L	L	H	H	L	L	H	H	H	H	H	H	**L**	H
L	H	L	H	L	L	H	H	H	H	H	**L**	H	H
L	H	H	H	L	L	H	H	H	H	**L**	H	H	H
H	L	L	H	L	L	H	H	H	**L**	H	H	H	H
H	L	H	H	L	L	H	H	**L**	H	H	H	H	H
H	H	L	H	L	L	H	**L**	H	H	H	H	H	H
H	H	H	H	L	L	**L**	H	H	H	H	H	H	H

Table 2. Truth-table for the 74HC138 decoder/demultiplexer.

functionality to the '374, but all the data inputs are on the left side of the chip and all the outputs are on the right side of the chip. This makes it much easier to wire up; many people commonly use the '574 for this reason, even though the two devices are logically identical.

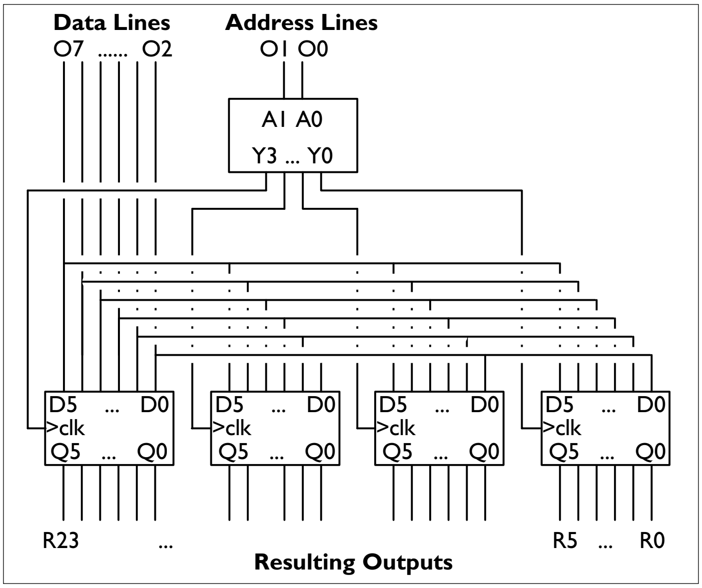
Putting It All Together

How does this work? Let's use an example to walk through the setup and use of a typical system.

The Hardware

Assume we have eight system outputs (O0 through O7) from a microcontroller and we want more outputs. Let's call two of these outputs address lines and six of them data lines. With two address lines, we will generate four control outputs (Y0 through Y3 in Figure 6). Each of these is connected to the clock input of one of the flip-flop devices. The six data lines (O2 through O7) are connected to the data input lines (D0 through D5) of all four of the hex flip-flops. The resulting outputs are generated from the four flip-flops as outputs R0 through R23. Thus, with five common off-the-shelf chips, we have turned eight outputs into 24.

Figure 6. Eight-bit to 24-bit multiplexer circuit.



The Software

To use the circuit detailed above, the software in the system must know about the specific hardware connected to the eight-bit output lines. Given a 24-bit number to output, what does the software need to do? Simply stated:

We need to present the correct data to the six data lines and then cause the correct output from the '138 to go from low to high to capture the data on the data lines into the correct latch.

Okay, we'll walk through this specific example step-by-step, but first, we need to delve into the topic of Gray codes. Frank Gray — a research scientist at Bell Labs — patented the Gray code encoding vacuum tube. We're not using vacuum tubes, but the principle is very important here, too. Gray code is a sequence of binary numbers such that any two consecutive numbers differ only in a single position. Let's consider the repeating cycle of numbers 0 through 3, as seen in Table 3.

The red cells in the table cause a problem in our decoding circuit; two bits are changing at the same time as we enter the cell. An actual digital circuit cannot change two things at exactly the same time; there will be a slight lag in one of them. So, in switching from binary number 00 to 11, for example, the circuit will momentarily pass through either 01 or 10. This will cause a very short change in the corresponding Y1 or Y2 output of the decoder, which will cause the wrong data to be captured in the correspon-ding latch. I've seen this problem in embedded software; it is hard to find and really messes things up! The key is to only change one bit at a time, even from the last value back to the first value, so the decoder will change its Y outputs in a very controlled manner, which is good. We don't need to change anything in the circuit to achieve this; we just need to remember to use Gray code switching in our controlling software or else our outputs will be very strange and not what we want them to be. You may think you have a hardware circuit error. Now, on to our code.

Starting assumption: The address bits on the output line are all low from the previous write.

1. Set the address lines O1-O0 to LL (should already be in this state).

2. Set the data values V5-V0 on data lines O7-O2.

3. Set the address lines O1-O0 to LH. This causes the data values to be captured in the first (rightmost)

Decimal	Binary	Gray Code
0	00	00
1	01	01
2	10	11
3	11	10
0	00	00
1	01	01
2	10	11
3	11	10

Table 3. Gray code for a two-bit number cycling from top to bottom.

flip-flop as line Y0 goes from LOW to HIGH. Also note that Y1 will now be LOW.

4. Set the data values V11-V6 on data lines O7-O2.

5. Set the address lines O1-O0 to HH. (Remember, we need to use Gray code!) This causes the data values to be captured in the second flip-flop.

6. Set the data values V23-V18 on data lines O7-O2.

7. Set the address lines O1-O0 to HL. This causes the data values to be captured in the forth flip-flop.

8. Set the data values V17-V12 on data lines O7-O2.

9. Set the address lines O1-O0 to LL. This causes the data values to be captured in the third flip-flop.

Of course, in your system, you would write a subroutine to do this, so your program would just invoke the subroutine to get the job done. What does this software look like? How does it do the eight steps detailed above?

Listing 1 (view it at **www.nutsvolts.com**) shows a detailed assembly language program for the 68HC11. The operations used are bit-level manipulations, such as bit-shifting, bit-masking, bit-setting, and bit-clearing. You can accomplish these operations using mathematic operations, but the clearest way to do these operations is to use the bit-level operations in your system. (I'll bet you always wondered what they were and why anyone would ever want to use them!) In assembly code, these operations are the logical shift and bitwise AND, OR, and NOT operators. In C, C++, and Java, look at the shift (**<< >>**) operations and the bitwise (**| &** ^) operators.

Practical Limitations

We've seen the theoretical fan-out we can achieve, but let's consider the common building blocks and what would be practical to build and use. We could use devices without using all of their capabilities, but — if you're building an external circuit for your system — you would want to keep

Starting Outputs	Address Lines (A)	Data Lines (D)	Total Outputs	74HC138 3-8 decoder	74HC74 2-bit FF	74HC173 4-bit FF	74HC174 6-bit FF	74HC374 8-bit FF
8	0	8	**8**	0	0	0	0	0
8	1	7	**14**	1	0	0	0	2
8	2	6	**24**	1	0	0	4	0
8	3	5	**40**	1	0	0	8	0
8	4	4	**64**	2	0	16	0	0
8	5	3	**96**	4	0	32	0	0
8	6	2	**128**	8	64	0	0	0
8	7	1	**128**	14	64	0	0	0

Table 4. Numbers of devices needed for circuits with various latch sizes.

it as small as possible for reasons of cost, footprint (size), and complexity to build.

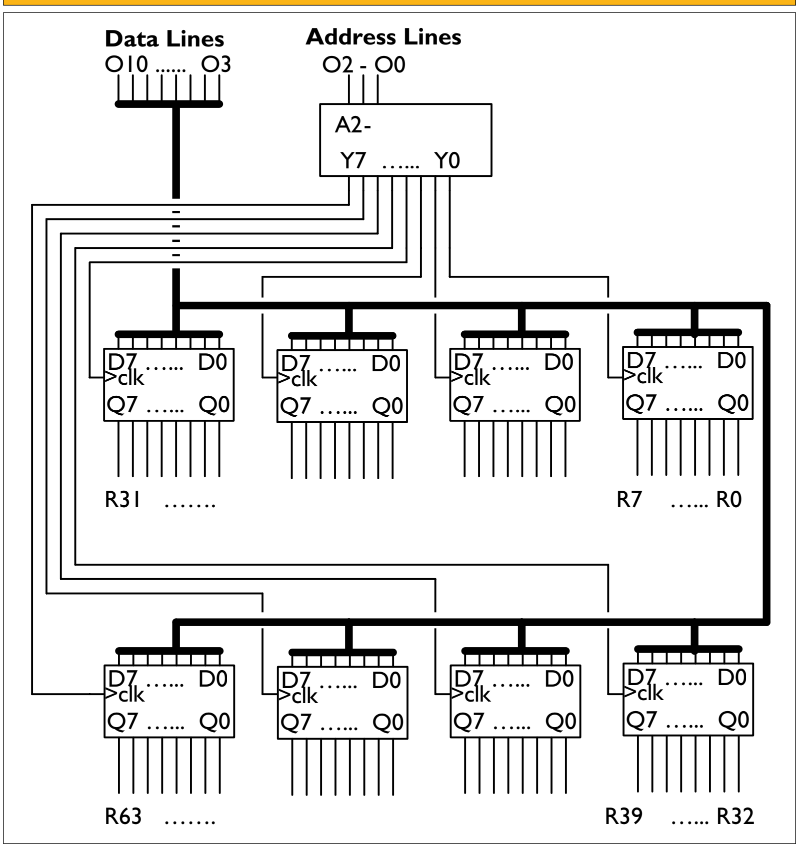
We see the 74HC138 decoder as a great one-chip workhorse that can directly take three address lines and produce eight control lines. Two '138 devices can be combined with four address lines. Three are in common and the fourth selects which '138 to use; for example, address lines A0-A2 are connected to the address inputs of both '138s and the last address line, A3, is connected to E1 on one '138 and E3 on the other. Two '138s will produce 16 control lines — or you could use a single 74HC154. The '138 is a 16-pin chip while the '154 is a wider, 24-pin chip. Multiple '138 chips are commonly used. A single '138 can be used with only two address lines (A0 and A1, connect A3 to ground), which would then only generate four usable control lines — Y0 through Y3.

The other type of device required is a flip-flop to capture the outputs. The four devices discussed handle two, four, six, and eight bits. The four-, six-, and eight-bit devices all have a common clock, which means that all bits will be changed at the same time. For some applications, this is a problem, but — for our purposes — this is what we want. Therefore, the "best" combination is one which uses six or eight data lines, since these can be captured in a single flip-flop device for each address. If you use five data bits, you would use a six-bit flip-flop and not use one of the bits and so on.

Let's revisit Table 1, but add columns for the number of chips required to implement each circuit. I've highlighted the interesting numbers in Table 4 in red, showing the number of decoders and flip-flops needed to implement the circuit.

Let's suppose you really need 128 outputs. The two solutions above can accomplish this, but need from 72 to 78 chips. I don't know about you, but I wouldn't want to build that! Let's consider different combinations of address and data lines (assuming you have a few more than eight starting system outputs to work with) to achieve 128 outputs. Note that some of these circuits will not use all control outputs generated. Again, I've highlighted the interesting numbers in red in Table 5.

Figure 7. An 11-bit to 64-bit multiplexer circuit.



We clearly see that, if we can spare a few more system outputs, we can easily generate 128 outputs with much less external circuitry (and effort). If I needed 128 outputs, I know I would try to come up with 12 starting outputs and build the 18-chip circuit.

I should mention that you could use a two-level fan-out scheme, such as using five starting outputs to fan-out to 12 outputs (two address lines and three data lines), then use these 12 intermediate outputs to fan-out to 128 outputs, as we saw above. The software would need to use four writes to generate a single 12-bit number and it would require 16 12-bit numbers to set all 128 output bits. There is also the issue of timing with the edge-triggered flip-flops, which would really force you to use six starting bits (two address lines and four data lines) to allow the four address lines for the second stage to be changed all at the same time. Using multi-level designs, you can

fan-out to as many outputs as desired with a very small number of starting outputs, but the required hardware and software would become more and more complex.

Starting Outputs	Address Lines (A)	Data Lines (D)	Total Outputs	74HC138 3-8 decoder	74HC74 2-bit FF	74HC173 4-bit FF	74HC174 6-bit FF	74HC374 8-bit FF
8	6	2	128	8	64	0	0	0
9	5	4	128	4	0	32	0	0
10	5	5	128	4	0	0	26	0
11	5	6	128	4	0	0	22	0
12	4	8	128	2	0	0	0	16

Table 5. Numbers of devices needed for circuits with various latch sizes to generate 128 outputs.

Example Circuits and Software

Let's choose a couple of desired configurations and look at the detailed circuits and software. Two that I'm currently adding to 68HC11 systems for robotic use will go from eight system outputs to 24 outputs and from 11 system outputs to 64 outputs. We will then go on to detail how to use these in the supporting software. Of course, my hope is that you can take the information presented here and customize it to build up the desired circuit and software for your particular application.

Eight Outputs to 24 Outputs

The program to drive the miltiplexer circuit is organized as a method that is called to set the entire 24-bit number. The application program just calls this method whenever it wants to change these outputs. The pseudo code for this is what we saw before. The hardware requires a two-bit address and a six-bit data value. Let's assume this is connected to the B port on the 68HC11, which resides at address $1004.

The software to do this in 68HC11 assembly language is shown in Software Listing 1 (available at **www.nutsvolts.com**). Anything after a ";" sign is a comment, used to describe the logic.

11 Outputs to 64 Outputs

Building on the ideas we've discussed, here is a nine-chip solution to expanding 11 system outputs (three address and eight data) to 64 outputs. It uses a single 74HC138 decoder and uses the eight lines from it to drive eight eight-bit latches for a total of 64 bits.

The circuit can be easily wired up from the connections in Figure 7 and the software will be an expansion of the code for the 10-to-32 multiplexer circuit in Software Listing 2 (view it at **www.nutsvolts.com**).

About the Author

*Tom Dickens is an engineer and Associate Technical Fellow at The Boeing Company. He maintains a website of 68HC11 information and examples at **http://tomdickens.com/***

Conclusions

We have discussed the technique of expanding a small number of digital outputs into a very large number of outputs. We have considered the available off-the-shelf devices commonly used to implement a multiplexer circuit, which, in turn, dictates the practical limits to these designs. **NV**

Standalone Watchdog Timers

An External Guard Circuit for Your µC Projects

What's a watchdog timer? Many of you who have worked with single board computers are familiar with watchdog timer circuits. These circuits monitor the operation of the microcontroller and its software and restart it if a problem is detected. Watchdog timers are an excellent way to insure that a microcontroller-based system continues to operate unattended if occasional failures occur.

There are a number of reasons why your microcontroller might need this kind of monitoring: operation in a harsh environment that creates electrical "spikes," hardware conditions that the software could not be tested for, and — yes — even "bugs" in the software.

The way that a watchdog timer does its job is to monitor a particular signal sent by the microcontroller. If this signal does not occur on a regular basis, it assumes the microcontroller is "locked up" or the software is "lost" and cannot send this signal. It then restarts the system.

What We Needed

I was talking to one of our customers recently about our RC51 Programmable Relay Controller, which is based on an Atmel AT89C4051 microcontroller chip. He asked about the reliability of microcontroller-based systems in harsh environments. Although our full-featured, single board computer has a built-in watchdog timer chip, our single microcontroller chip-based products — like the RC51 — do not.

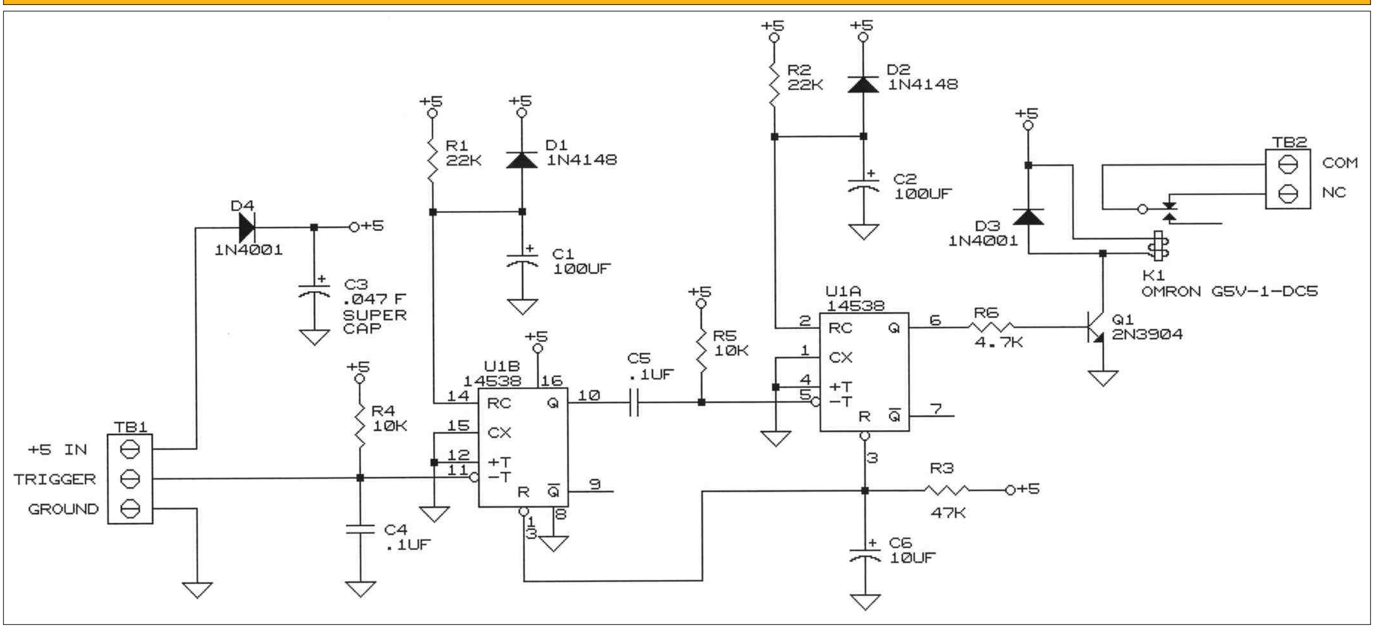
Even though we have found that the RC51 and other products based on single microcontroller chips are very resistant to the problems associated with harsh environments, I realized that it would be nice to increase its reliability further — if the need arose — with a simple standalone watchdog timer circuit.

What I Designed

Watchdog timer circuits can either be included in the circuitry on a single board computer or microcontroller or they can be standalone devices connected to a variety of signals. If the circuit is included on a single board computer, it usually restarts the software by issuing a hardware reset to the microcontroller, just as though someone had pressed a reset button.

What if your microcontroller does not already have a watchdog circuit? Any add-on circuit would need to interface to the reset circuitry on the board. This may

Figure 1. Schematic of the watchdog circuit. Note that C3 is in Farads, not µF.



not be a practical thing to do, but, if the watchdog timer is a standalone circuit with a relay, it can cycle the power so that the microcontroller will reset itself when powered up.

Since all of our controllers (and most single board computers) have a source of 5 volts to power small external circuits, in addition to an available spare logic level output, I was set. I decided to design a simple circuit that is easy to connect to nearly any small, single board computer or microcontroller circuit that can restart these devices by power cycling them.

Circuit Description

The circuit shown in Figure 1 is based on a common CMOS chip — the MC14538 — or equivalent dual edge triggerable, retriggerable, and resettable monostable multivibrator (timer). Its ability to lengthen its output pulse when retriggered is the basis for the operation of this circuit. Although the 14538 is versatile enough to have both rising and falling edge trigger inputs, only the falling edge triggers are used.

The first timer section is used to monitor a low going logic level or current sink trigger pulse, which the microcontroller being monitored must continue to pulse at a certain interval. The second timer section is used to activate the relay for a specified amount of time once the first timer determines that the trigger pulse has stopped. (With the values listed in the Parts List, both of these time periods are set to approximately 2.2 seconds.)

When you are ready to begin using the watchdog timer circuit, all you need to do is have the microcontroller begin sending low going pulses to it at least once every second or so. From that point on, the microcontroller must continue to send these pulses or the relay will activate and restart the microprocessor. The values shown in the schematic and Parts List provide a time of about 2 seconds for each timer section and can be changed, as described in the next section.

The relay was chosen for its small size and fairly low coil current consumption — about 30 mA at 5 volts. Its contacts can handle 1 amp at 24 volts — more than enough to cycle the power of many microcontroller circuits and boards. Another component that deserves mention is the .047 farad, 5.5 V "SuperCap" used for C3. This capacitor is a Panasonic EEC-F5R5U473 or equivalent and was chosen over electrolytic capacitors because of its large capacitance for its size.

Parts List

U1	MC14538
K1	Omron G5V-1-DC5 (see text)
C1,C2	100 µF electrolytic
C3	.047 Farad, 5.5 V "SuperCap"
C4,C5	0.1 µF
C6	10 µF electrolytic
D1,D2	1N4148 or 1N914
D3,D4	1N4001
Q1	2N3904 or 2N2222
R1,R2	100K 1/4 W
R3	47K 1/4 W
R4,R5	10K 1/4 W
R6	4.7K 1/4 W
TB1	3 pin terminal block
TB2	2 pin terminal block

Alternate parts

D4	1N5819 Schottkey diode
U1	74HC4538N

How It Works

D4 isolates the 5 volt power going to the watchdog circuit. If the watchdog timer removes power from the microcontroller circuit (which is providing the watchdog with power), the charge in C3 will be blocked and will not try to go back and power the microcontroller.

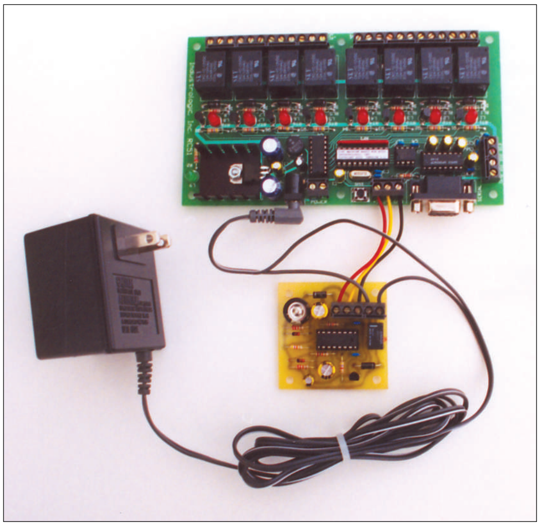
Since the circuit uses such a small amount of current, the voltage drop across D4 is quite small. The measured voltage with most 1N4001 series diodes is 4.7 volts or more when C3 is fully charged. This voltage meets the requirements for CMOS chips (3 to 18 volts) and is more than enough to operate the relay, which has a coil voltage rating of 5 volts and a pull-in voltage of 80% of its rated voltage.

C6 and R3 form a power-up reset to the chip's reset pins on both sections to insure that the outputs are in the reset state ("Q" outputs low) after the power stabilizes upon power-up. D1 and D2 are recommended by the chip's data sheet to avoid large discharge currents through the chip when large value capacitors are used for long timing delays.

R4 provides a pullup for the trigger input of the first timer so that an open collector signal, as well as a logic level signal, can be used to pull it low and provide the falling edge to trigger it. C4 is a filter and provides a small degree of protection from static and false triggering.

When a low going trigger pulse is sent from the micro-controller, the first timer will start and will set its Q output high. Q will remain high as long as another pulse occurs before the timing period ends; otherwise, it will return low. C1 and R1 control the time period of the first timer. R5 holds the falling edge trigger of the second timer high until C5 pulls it low through the Q output of the first

Figure 2. The circuit (bottom) guarding a controller.



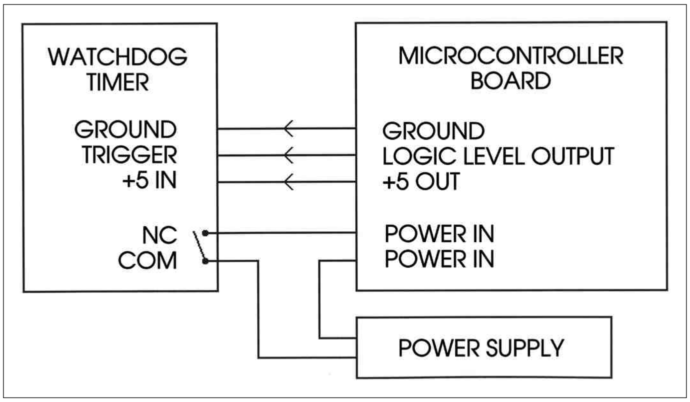


Figure 3. The system wiring diagram.

timer. This arrangement creates a falling edge before the capacitor charges and allows only a high to low transition of the Q output to trigger the second timer that controls the relay.

When the second timer is triggered, its Q output turns on the relay via R6 and Q1 by sinking the relay current to ground. C2 and R2 control the time period of the second timer, which is the relay "on time."

When the relay activates, its normally closed contact that supplies power to the microcontroller being monitored opens and removes the power. This, of course, removes power from the watchdog circuit, but C3 has sufficient charge to operate the relay for a short period of time. When the relay deactivates, power is returned to the microcontroller (which restarts) and to the watchdog timer circuit.

Building the Circuit

For prototypes of simple circuits that can be laid out on single-sided printed circuit boards (PCBs), I prefer to etch my own handmade boards. Circuit layout is done with PCB layout software and then printed at a one-to-one scale to use as a guide to drilling the holes. A larger version is then printed in reverse to be used to create the layout with dry transfer patterns and a resist pen. (Actually, I often use diluted fingernail polish and a paintbrush with only 10 hairs remaining to paint on the circuit paths.) Figure 2 shows a picture of the completed board connected to our RC51 Programmable Relay Controller.

Using the Watchdog Timer

Figure 3 is a block diagram of how the watchdog timer is connected to the RC51 and other microcontroller boards. The digital output signal used during testing with the RC51 was the general-purpose digital I/O signal called “INT.” The RC51 has a simple onboard language called Tiny Machine Basic, so it was easy to write a test program that shows how to trigger the watchdog timer. The program below allows the RC51 to be used as an “RS-232 relay” board where a host computer can operate relays based on the binary value of a character sent to the RC51.

```
1 INT=0     set the digital output low (trigger watch-
dog)
2 INT=1     set the digital output back high
3 A=KEY     get the character at the serial port
4 IF A=0 1 if there is no character, keep looping
5 RELAYS=A set relays to binary value of character
6 GOTO 1    keep looping
```

Modifications to the Circuit

C1 and R1 control the period of the first timer. This time determines how long the circuit will wait for a pulse from the microcontroller before determining that there is a problem. C2 and R2 control the relay "on time" and can be changed. (Remember, however, that C3 may not have enough charge to provide power to the circuit for long periods of time.) The time delays for both timer sections are calculated like a simple R/C time constant, that is, capacitance in farads (microfarads divided by 1,000,000) times resistance in Ω. As stated, a Schottky diode like the 1N5819 can be used for D4 instead of the 1N4001. This diode has a lower voltage drop and the measured voltage of the circuit will be very close to 5 volts. This will be helpful if other relays are selected that use more current.

A 74HC4538 high speed CMOS chip can also be used instead of the CMOS 14538, in which case the Schottky diode may be desired, since the HC part uses slightly more current. If your single board computer has a reset switch that terminates at a connector, you may want the watchdog timer to restart your microcontroller with a reset. In that case, you can use the normally open contact on the relay instead of the normally closed contact, so that the relay acts like a reset switch. You may also want to shorten the time delay on the second timer section by using a lower value for R1.

The watchdog timer circuit can, of course, be powered by its own 5 volt power supply. This will allow the second timer to complete its entire timing cycle set by the timing components R2 and C2, rather than ending prematurely when the power is removed. This modification may be required if there is a specific power off time needed to restart the system. **NV**

About the Author

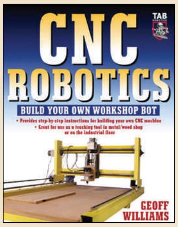
Gary Peek is the President and co-founder of Industologic, Inc., a manufacturer of microcontroller-based industrial data acquisition and control products. He can be contacted at peek@industrologic.com

Selected Titles for the Electronics Hobbyist and Technician—

The *Nuts & Volts* Hobbyist Bookstore

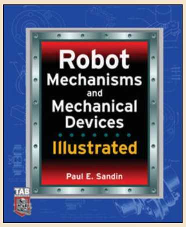
Robotics

CNC Robotics
by Geoff Williams



Written by an accomplished workshop bot designer/builder, *CNC Robotics* gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80% of the price of an off-the-shelf bot — and can be customized to suit your purposes exactly because you designed it. **$34.95**

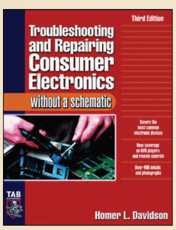
Robot Mechanisms and Mechanical Devices Illustrated
by Paul Sandin



Both hobbyists and professionals will treasure this unique and distinctive sourcebook — the most thorough and thoroughly explained compendium of robot mechanisms and devices ever assembled. Written and illustrated specifically for people fascinated with mobile robots, *Robot Mechanisms and Mechanical Devices Illustrated* offers a one-stop source for everything needed for the mechanical design of state-of-the-art mobile bots. Written by a leading designer of robots used at the horizon of mobile robotics, this resource offers a collection of both new and classic robotic mechanisms and devices unmatched in scope — from such high-level sources as the mechanical engineers' mainstay, *Mechanisms and Mechanical Devices Illustrated*. Paul Sandin's superlative reference also brings you new robotic mechanisms and devices that have never before been collected! **$39.95**

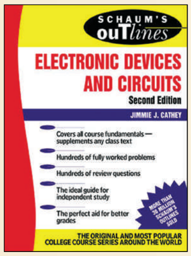
Electronics

Troubleshooting & Repairing Consumer Electronics Without a Schematic
by Homer Davidson



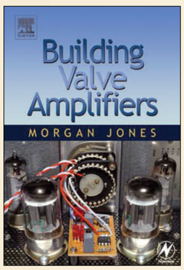
In this book, Homer Davidson gives you hands-on, illustrated guidance on how to troubleshoot and repair a wide range of electronic products — when you can't get your hands on the schematic diagrams. He shows you how to diagnose and solve circuit and mechanical problems in car stereos, cassette players, CD players, VCRs, TVs and TV/VCR combos, DVD players, power supplies, remote controls, and more. **$34.95**

Schaum's Outline of Electronic Devices and Circuits, Second Edition
by Jim Cathey



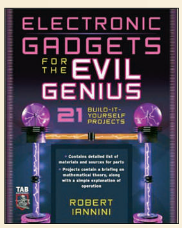
This updated version of its internationally popular predecessor provides introductory problem-solving text for understanding fundamental concepts of electronic devices, their design, and their circuitry. In addition to providing an interface with Pspice (the most widely used program in electronics), new key features include a new chapter presenting the basics of switched mode power supplies, 31 new examples, and 23 PS solved problems. **$16.95**

Building Valve Amplifiers
by Morgan Jones



Building Valve Amplifiers is a unique, hands-on guide for anyone working with tube audio equipment — as an electronics experimenter, audiophile, or audio engineer. Particular attention has been paid to answering questions commonly asked by newcomers to the world of the vacuum tube, whether it's audio enthusiasts tackling their first build or more experienced amplifier designers seeking to learn the ropes of working with valves. The practical side of this book is reinforced by the many clear illustrations throughout. **$29.99**

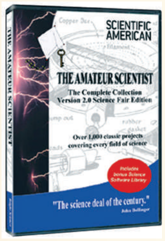
Electronic Gadgets for the Evil Genius
by Robert Iannini



The do-it-yourself hobbyist market — particularly in the area of electronics — is hotter than ever. This books gives the "evil genius" loads of projects to delve into — from an ultrasonic microphone to a body heat detector, all the way to a *Star Wars* Light Saber. This book makes creating these devices fun, inexpensive, and easy. **$24.95**

WE ACCEPT VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.
Ask about our 10% subscriber discount on selected titles.

The Amateur Scientist 2.0 Science Fair Edition
from "The Amateur Scientist" column

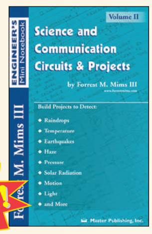


This CD contains the complete collection — 73 years — of articles from *Scientific American Magazine*'s legendary column "The Amateur Scientist," plus a second Science Software Library CD with dozens of shareware and free-ware programs to feed the passion of any science nut. With over 1,100 projects to challenge science enthusiasts of all ages and skill levels — rated by cost, potential hazard, and difficulty — this is the ultimate resource for anyone interested in home-based science. If that's not enough, it also contains over 1,000 bonus pages of additional how-to science techniques that never appeared in *Scientific American*. Great for science fair students, hobbyists of all ages, and home-schoolers! In fact, *The Amateur Scientist 2.0* contains a special primer for science fair students. Fully text-searchable and packaged in an attractive double-CD case, this remarkable browser-based product runs seamlessly on every platform — Windows, Macintosh, Linux, and Unix.

$24.99 — Subscriber
$29.99 — Non-subscriber

Engineer's Mini Notebook Collection
by Forrest M. Mims III

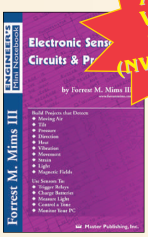
Volume 1 features more than two dozen 555 timer circuits that you can build. **$10.95**



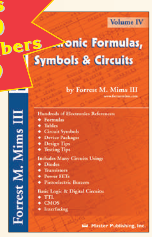
Volume 2 — Study rain, lightning, clouds, sunlight, water, temperature, and much more! **$10.95**

NEW!

All Four Volumes $39.99 (NV subscribers ONLY!)



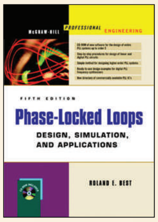
Volume 3 — Learn about important sensors and use them to build circuits and projects. **$10.95**



Volume 4 includes frequently used electronic formulas, tables, circuit symbols, and more! **$10.95**

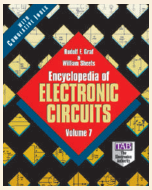
Call 1-800-783-4624 today! Or order online at www.nutsvolts.com

Phase-Locked Loops
by Roland Best



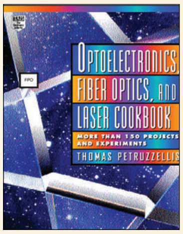
The fifth edition of this classic circuit reference comes complete with extremely valuable PLL design software written by Dr. Best. The software alone is worth many times the price of the book. The new edition also includes new chapters on frequency synthesis, CAD for PLLs, mixed-signal PLLs, and a completely new collection of sample communications applications. **$79.95**

Encyclopedia of Electronic Circuits, Volume Seven
by Rudy Graf



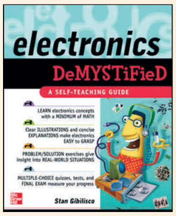
Designed for quick reference and on-the-job use, the *Encyclopedia of Electronic Circuits*, Volume Seven, puts over 1,000 state-of-the-art electronic and integrated circuit designs at your fingertips. This collection includes the latest designs from industry giants, such as Advanced Micro Devices, Motorola, Teledyne, GE, and others, as well as your favorite publications, including *Nuts & Volts*! **$39.95**

Optoelectronics, Fiber Optics, and Laser Cookbook
by Thomas Petruzzellis



This is a practical guide to one of the hottest fields in electronics and optical circuits. A collection of hands-on experiments and projects for the student, technician, and hobbyist, it explains optoelectronics in nontechnical terms. Projects show how optical circuits work and how to use them in practical and efficient ways. You'll save time, money, and energy with dozens of do-it-yourself projects — from laser alarm systems to high-speed fiberoptic data links. Circuit diagrams, schematics, and complete parts lists accompany each project and an appendix lists suppliers for needed parts. **$29.95**

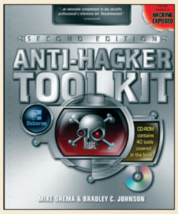
Electronics Demystified
by Stan Gibilisco



Best selling *Demystified* author and electronics expert Stan Gibilisco has penned the perfect introductory book for consumers, hobbyists, and students alike. Coverage includes essential topics, such as current and power supplies, wireless, digital principles, measurement and monitoring, transducers and sensors, location and navigation, and more. **$19.95**

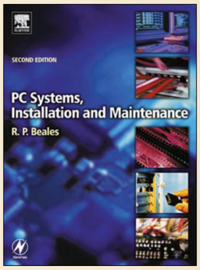
Home Computing

Anti-Hacker Tool Kit, Second Edition
by Mike Shema / Brad Johnson



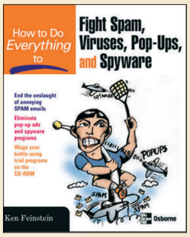
Get in-depth details on the most effective security tools and learn how to use them with this hands-on resource. A must have companion to the bestselling security book *Hacking Exposed*, this tool kit includes tips and configuration advice for getting the best results from the top hacking tools created and in use today. **$59.99**

PC Systems, Installation and Maintenance, Second Edition
by R. P. Beales



Written in a straightforward, easy-to-read style, Rob Beales provides the knowledge and techniques needed to build, troubleshoot, and maintain personal computer systems. Case studies and practical working examples are included throughout the text, with additional case studies specifically aimed to meet the requirements of e-Quals courses on an accompanying website. Further web resources include key figures from the text available to download in full-color, with a wealth of extra material covering Binary/Hex and basic logic functions, ASCII tables, connector types and pinouts, bus slots, RAM slots, and further useful website links. **$29.99**

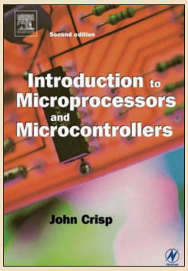
How to Do Everything to Fight Spam, Viruses, Pop-Ups, and Spyware
by Ken Feinstein



Swat spam, vanquish viruses, knock down pop-ups, and expose spyware with help from this one-of-a-kind resource. Loaded with insightful advice, practical tips, and trial software, this book-and-CD defense will help you rid your computer of the perils and nuisances of web surfing once and for all. Author and tech expert Ken Feinstein gives easy-to-understand explanations of the technologies at work and just exactly what steps you can take to take back your Email, your surfing enjoyment, your privacy, and your computer. **$24.99**

Microcontrollers

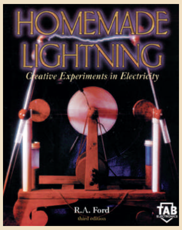
Introduction to Microprocessors and Microcontrollers
by John Crisp



Assuming only a general science education, this book introduces the workings of the microprocessor, its applications, and programming in assembler and high level languages, such as C and Java. Practical work and knowledge-check questions contribute to building a thorough understanding with a practical focus. This book concludes with a step-by-step walk through a project based on the PIC microcontroller. The concise but clearly written text makes this an ideal book for electronics and IT students and a wide range of technicians and engineers, including IT systems support staff and maintenance/service engineers. **$34.99**

High Voltage

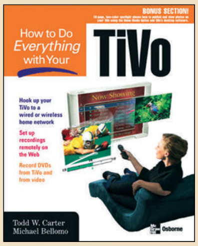
Homemade Lightning: Creative Experiments in Electricity
by R. A. Ford



Enter the wide-open frontier of high-voltage electrostatics with this fascinating, experiment-filled guide. You'll discover how to make your own equipment, how electricity is used in healing, and how experiments in high potential physics work! **$24.95**

Home Entertainment

How to Do Everything with Your TiVo
by Todd Carter



Turn on your TV! This helpful resource helps you select the best TiVo service based on your needs, then shows you how to set up your TiVo, watch live television while your it, record programs, hook TiVo up to your home network, remotely schedule programs to be recorded over the Internet, use multiple TiVos, and much more! **$24.99**

If you don't see what you need here, check out our online store at **www.nutsvolts.com** for a complete listing of the titles available.

DIGITAL CONTROL Without Programming

by David Caldwell

A dedicated digital controller with intuitive settings makes it quick and easy to close a loop around a power supply, heater, motor, lamp, or other device.

Closed-Loop Challenges

Closed-loop controllers make automatic adjustments to maintain constant output despite varying conditions. Examples include supply voltage, fluid temperature, motor speed, and light intensity. These parameters would shift over time and load without consistent correction.

Microcontrollers executing firmware are replacing op-amps with feedback networks that used to perform closed-loop control. These digital implementations are versatile but time-consuming to program and control demands often exceed processor resources. Considerable expertise is required to properly design a system to avoid oscillations or sluggish responses. We naturally observe events in the time-domain while control analysis is typically done in the frequency-domain, which can be complex and confusing.

One solution to these challenges is an automated closed-loop controller configured by intuitive time-domain settings. CLOZD™ is a control chip developed by Flextek Electronics that is versatile and easy to use. Just select the desired timing configuration through pin settings and quickly close a loop around a power supply, motor drive, lamp, heater, fan, Peltier, valve, actuator, or amplifier.

Controller drive is automatically adjusted until the measured sensor signal matches the desired command, as shown in Figure 1. The Digital-Signal-Processing (DSP) algorithms within the CLZD010 control chip compare the feedback sensor signal and the analog setpoint command to determine appropriate Pulse-Width-Modulator (PWM) drive for the plant. Typical plants include power, thermal, motion, lighting, and flow applications.

Figure 1. CLOZD loop controller system application.

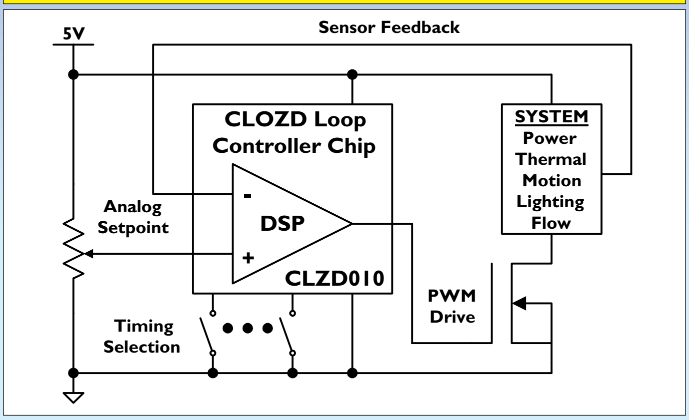
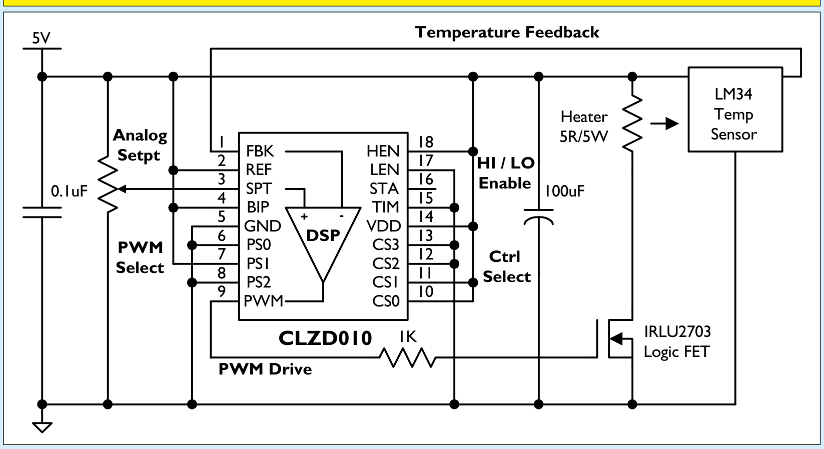


Figure 2. A thermal controller (for a 5 W heater).
PWM = 488 Hz (PS=010, BIP=1) CLOZD = 134S (CS=0011, TIM=0).



Thermal Controller

Thermal control systems are challenging because they have low level signals, long time constants, and multiple lag elements that can cause overshoot. However, the one shown in Figure 2 is quick and easy to configure for high performance with a few inexpensive parts. The duty cycle (percentage time

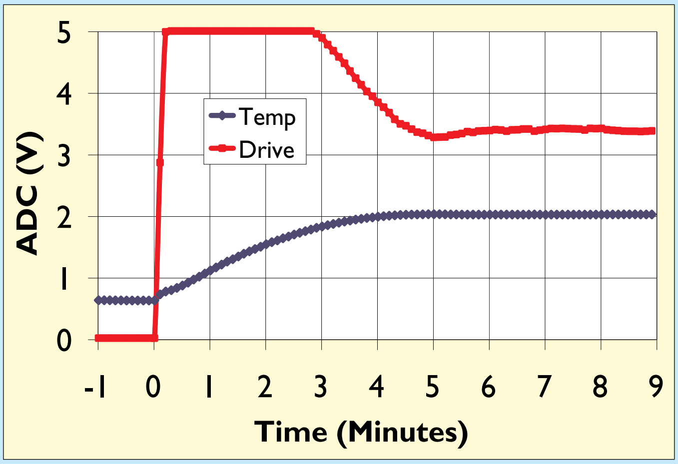


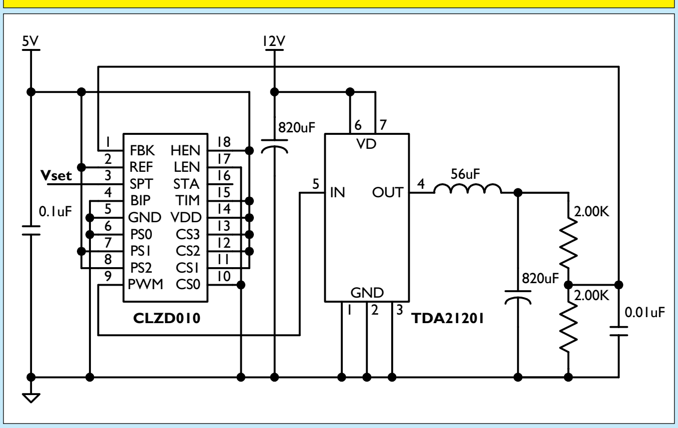
Figure 3. Temperature feedback and filtered PWM drive.

conducting) of the FET switch is adjusted until temperature feedback from the LM34 sensor is equal to the desired setpoint of the potentiometer. PWM frequency is set by the state of pins PS2-PS0 on CLZD010 while loop timing is set by pins CS3-CS0. Pins HEN and LEN are high and low enables.

The logic FET can be driven directly because low frequency PWM (PS2-PS0=488 Hz) is used with slow transitions (1K Ω gate resistor). Control timing was estimated by applying power to the heater and monitoring the temperature response. It took over 10 minutes for temperature to settle near its final value in an open-loop configuration. The temperature went from ambient to about two thirds of its final value in two to three minutes ($\tau = 1\text{-}e^{-1} \approx 63\%$), so the timing of the system was set slightly faster (CS3-CS0=134 S).

Figure 3 shows the filtered (average) PWM drive and the temperature FBK response as a function of time. Notice that drive is high while the loop error (FPT-FBK) is large, but decreases prior to the temperature reaching its final value of 200°F (FBK=2 V) for fast response without overshoot.

Figure 4. A switching power amplifier (10 V/10 A). PWM = 31.2 KHz (PS=110, BIP=0) CLOZD = 128 mS (CS=1110, TIM=1).



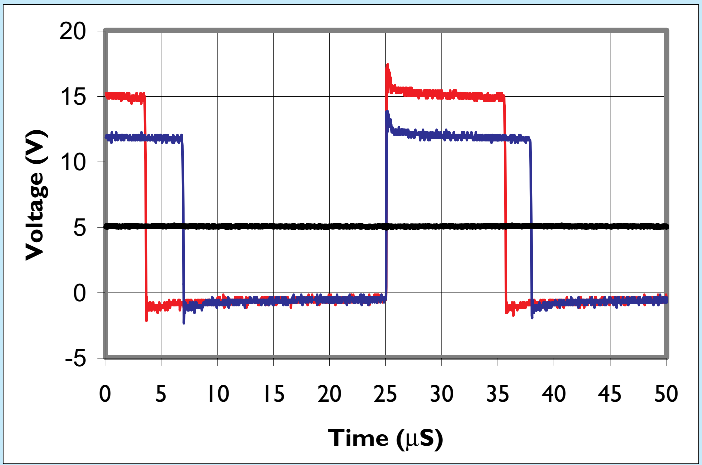


Figure 5. PWM drive waveforms for constant 5 V output with 12 V and 15 V supplies.

Power Amplifier

Figure 4 illustrates a power amplifier that sources or sinks current while maintaining constant output voltage at twice (1+2 K/2 K) the analog setpoint (Vset). This circuit is a switching converter that behaves like a low frequency (100 Hz) high power (10 V/5 A) op-amp. It requires few parts that are inherently robust because critical functions are integrated, including digital signal processing, power switching, and thermal shutdown.

The half-bridge driver TDA21201 converts PWM logic levels to a high power pulse train that is averaged to a DC level by the output LC filter, as shown in Figure 5. The LC response is fast and susceptible to ringing, so high speed sampling and control calculations are required. For this reason, the 128 µS control setting is used despite the 214 µS time constant of the system ($\tau = LC^{1/2}$).

Figure 5 illustrates that more drive (higher duty cycle) is required at a lower supply voltage (12 V versus 15 V) to maintain constant output voltage.

Bidirectional Controller

Figure 6 illustrates a thermal controller that uses a Thermo-Electric-Cooler (TEC) or Peltier Cell to heat (PWM>50%) or cool (PWM<50%), depending on current direction through full-bridge driver LMD18201. This circuit is useful for applications requiring variable temperatures that include ambient. The full-bridge has two outputs that switch out-of-phase; one is low while the other is high. Both filtered outputs are equal at half the input voltage when

PWM=50%, so no power passes through the TEC. By tying the bipolar BIP pin of the controller high, the PWM initializes at 50% duty cycle rather than the usual 0%.

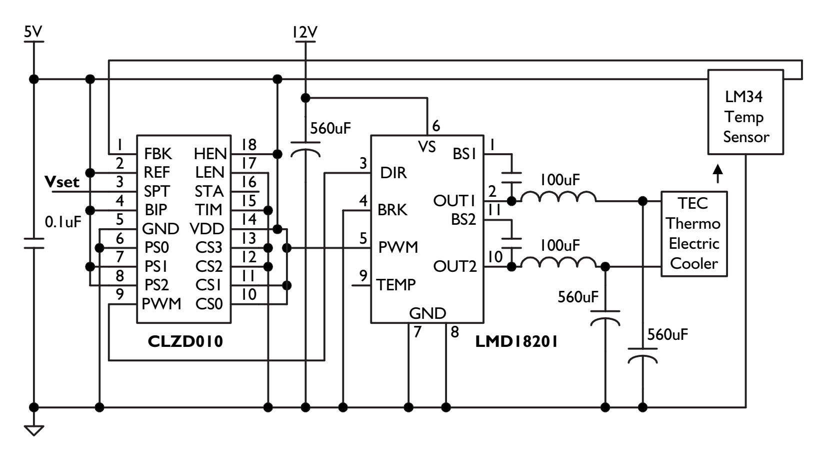


Figure 6. A bipolar thermo-electric cooler (12 V/3 A Peltier cell). PWM = 31.2 KHz (PS=110, BIP=1) CLOZD = 134 S (CS=0011, TIM=0).

PC USB Control

Two chips under $10.00 — each with free software drivers — enable the PC-based controller in Figure 7. The USB to serial converter provides the PC interface to the FlexController™ System-On-Chip (SOC), which commands setpoint and records system response through Visual Basic, while real-time control is performed by the CLOZD chip.

The FT232BM chip includes the hardware and PC drivers to communicate with microcontrollers through a USB port. The FCIC010 FlexController SOC combines the peripherals of a microcontroller with the ease of Visual Basic (or C++) programming through FTview™ Active X Control. These devices save valuable time by eliminating tedious low level programming and enabling customization through intuitive graphical development software.

Visual Basic code for control from PC USB:

```
Private Sub FTview1_NewData()
 FTview1.PWMduty = Setpoint        ' Write duty cycle
 Feedback = FTview1.AD0volt        ' Read AD0 voltage
 Write #1, Time, Setpoint,Feedback ' Save data to File
 FTview1.UpdateCmd = True          ' Update Control SOC
End Sub
```

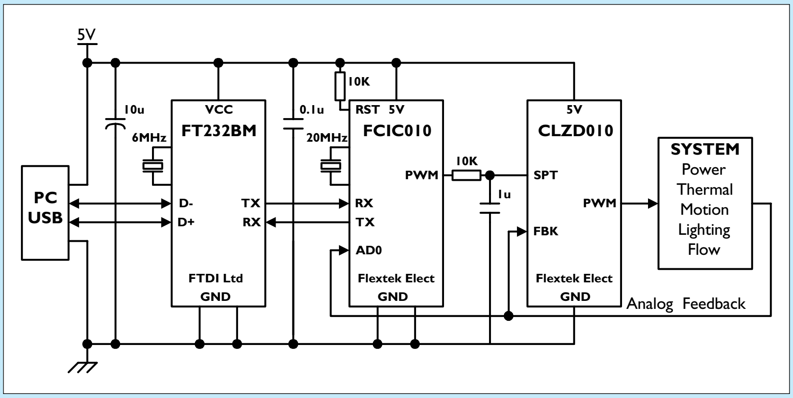


Figure 7. Closed-loop controller with PC USB interface.

Automated Control Advantages

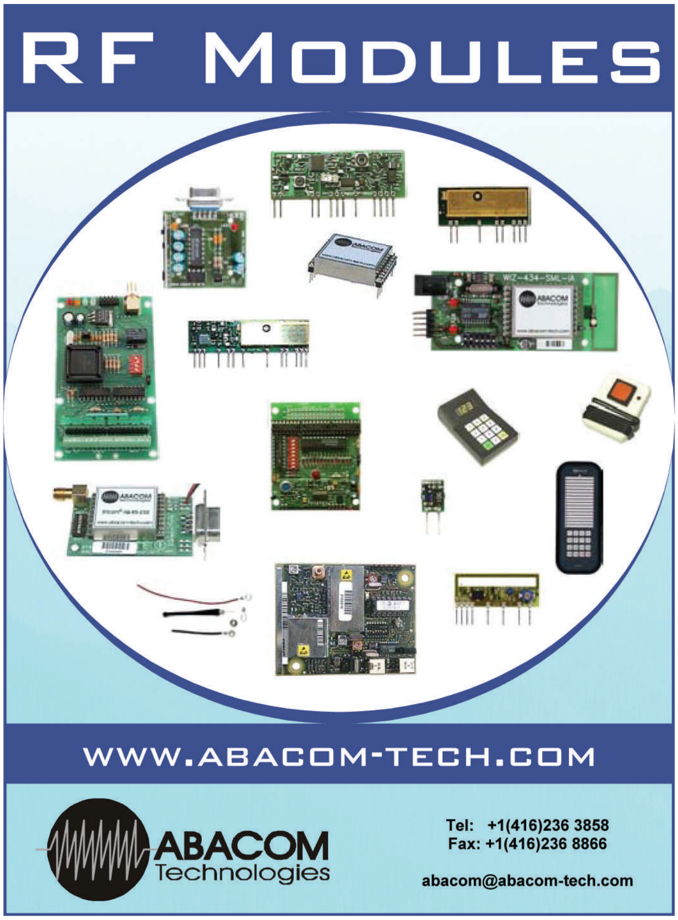
Closed loop control may be the most common and, yet, the most challenging task in electronics. By automating this task with versatile interfaces that are easy to use, a wide range of custom applications are quickly satisfied. Users save time and money by quickly configuring the same components for multiple applications. This allows savings on parts by purchasing in volume, reduced development time by reapplying familiar technology, and increased reliability by using proven components.

CLOZD Loop Controller utilizes advanced DSP techniques and algorithms to satisfy challenging real time control applications quickly and easily. Other methods and products require extensive programming and complex frequency-domain analysis. Configure the controller for a broad range of power, thermal, motion, lighting, and flow applications by selecting appropriate pin settings.

The CLSD010 control chip may be purchased for $5.95 with complete data sheet and application notes from **www.flex-tek.com** Other components referenced in this article may be purchased from **www.digi-key.com**, except the Peltier, which is available from **www.allelectronics.com** **NV**

About the Author

David Caldwell is the founder of Flextek Electronics, and has authored over a dozen papers and articles on power and control. You can reach him at djcaldwell@flex-tek.com



THE TRANSISTOR RADIO TURNS 50

A Look Back at an American Invention

by Sarah Lowrey

Fifty years ago this October, the world was a far different place than it is now. In 1954, almost anything electronic required vacuum tubes — an invention dating back to the beginning of the 20th century that had yet to be improved on.

Vacuum tubes were the only way of performing many electronic functions — such as amplification and rectification (although the solid-state diode had appeared on the scene by this time) — but were generally limited to devices that could be plugged in. As vacuum tubes required heating by filaments to function, the current use was high.

In addition, most standard vacuum tubes were large in comparison to other electronic devices. The combination of high power use and large size meant that few practical, portable devices utilizing vacuum tubes were created. Portable, battery powered radios had been around since the 1920s, but were limited by the requirement for several battery voltages, very short life of the batteries, expense of battery operation, and size of the vacuum tubes required. As a result, few portable radios saw much use.

The Transistor Arrives

The transistor — a revolutionary solid-state device capable of amplification — had been invented back in 1947, but its actual application to mass-produced consumer items had been limited, as production of transistors had yet to be perfected, so their cost remained high.

Transistors offered many improvements over the vacuum tube: They were smaller, used much less power, and were more reliable. Efforts were underway to refine transistor production so that individual devices would become inexpensive enough for use in consumer items.

The First Transistor Radio

By early 1954, Texas Instruments (TI) had perfected production to the point that transistors became cheap enough for use in consumer items. TI decided that a portable, handheld radio offered the most mainstream application of the new technology and approached several large corporations about producing the radio they had designed using TI transistors, but companies — such as RCA and Motorola — did not believe the transistor's time had yet arrived and passed on the offer. TI finally found a partner in a company called I.D.E.A., Inc., of Indianapolis, IN. Its main product up to that time had been a line of vacuum tube-operated television signal boosters marketed under the Regency brand name.

Figure 1. The Regency TR-1 was originally available in six standard colors. From left to right, top to bottom: black, white, cloud gray, mottled mahogany, Mandarin red, and jade green.

Figure 2. Early TR-1s had no coin slot to open them with.

I.D.E.A. jumped at the chance to produce a transistor radio. The target was to have the new radio on the market in time for the 1954 Christmas shopping season. As this was only weeks away, manufacturing the Regency-branded radio had to be a true rush job to meet the deadline.

Engineers decided upon a case size for the new radio. Attempts were made to refine the five-transistor circuit designed by TI to reduce cost and ensure that the parts would all fit in the case. One transistor was removed from the circuit and — after much effort — engineers succeeded in getting all parts to fit the case. An exception was a setscrew on the variable capacitor used for tuning — it stuck out far enough that a dimple had to be machined into the case to allow the back of the radio to close fully.

The result was the Regency TR-1 and it was introduced the week of October 18, 1954, to much fanfare and press. The radio's sleek design was due more to expediency and the desire to reduce cost rather than styling concerns. After considering various speaker grille options, the decision was made to simply drill holes in the case.

About the Regency TR-1

The TR-1 cost $49.95 — a princely sum back then. It utilized a 22.5 V battery, which — to my knowledge — was the only fully transistorized radio to ever use this battery. The radio came in six standard colors: black, white, cloud gray, Mandarin red, mottled mahogany, and jade green (see Figure 1). A special run of so-called "pearlescent" colors was introduced at a $5.00 premium and was quickly dropped. Apparently, consumers didn't think spending an extra $5.00 to get a special color was worth it. These pearlescent colors are extremely rare today and are sought after by collectors.

Unlike later radios, the TR-1 did not have a coin slot for opening the case at the bottom (see Figure 2). As a result,

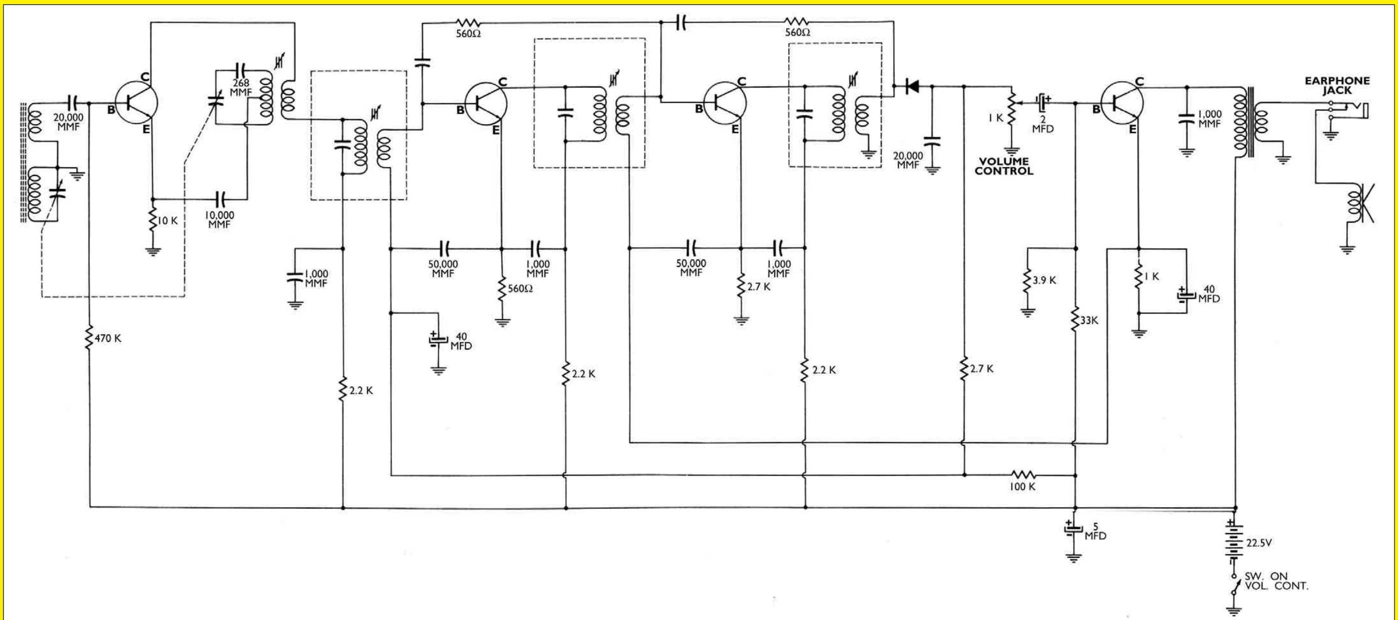


Figure 3. The original circuit drawing, as shown on the patent application. Four NPN transistors were used. Note that MMF = pF.

many TR-1s were damaged inadvertently when collectors attempted to open them the way they would any other radio, potentially cracking the case. The correct way to open the TR-1 is to gently squeeze each side of the case and remove the back.

Available accessories included a leather case and an

Figure 4.The TR-1 chassis, showing the tuning capacitor at the upper left, earphone jack at the upper right, transistors and transformer cans in the center, with the open frame speaker and battery clip toward the bottom.

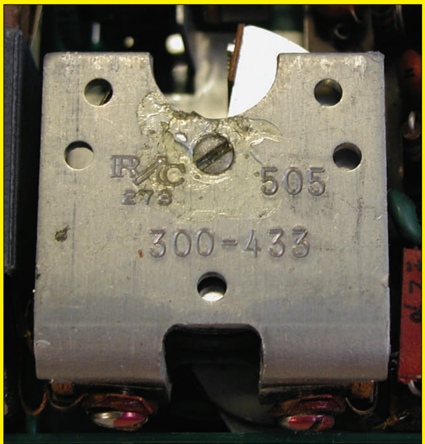


Figure 5.The date code for the capacitor is the number at upper right.

earphone. Unlike later radios, these accessories were extra cost options, with the case retailing for $3.95 and the earphone for an astounding $7.50!

The radio was designed using components of the day. The circuit is quite simple (see Figure 3), but it took up a lot of room, as the components were not the miniaturized ones we expect today. Figure 4 shows a view of the chassis. A large, open air tuning capacitor is visible, as are the large, open frame speakers and various transformers.

Note the oval-cased transistors. These early NPN transistors were germanium and of an early design known as "point contact." Each one is color-coded by type as to where it was to be installed. You can also see the setscrew on the tuning capacitor that required the dimple in the case.

Dating TR-1 Manufacture

The tuning capacitor can be used to approximately date the construction of each TR-1. Stamped into the back of the capacitor is a three-digit number that states the week and year that the capacitor was made (see Figure 5). It can be assumed that the capacitor was assembled into the radio shortly afterward. The number is at the top right and, in our example, is "505." The first number is the year, and the following two numbers are the week of the year. So, 505 translates to the fifth week in 1955 or about four months after the introduction of the TR-1. The serial number shown in Figure 4 reveals that almost 60,000 TR-1s had been manufactured by the spring of 1955.

When released, the TR-1 came in a simple yellow and black box. Regency quickly realized that the box could be used to advertise attributes of the radio and a new box was quickly designed (see Figure 6). Meant to be displayed on a counter, the radio was nestled in the box surrounded by the words "NO TUBES — ALL TRANSISTOR." Other marketing ploys included manufacturing TR-1 radios with clear backs to demonstrate the solid-state nature of the radio (see Figure 7). A few completely clear models were also made. These demonstration models are extremely rare today.

TR-1 "Clones"

The TR-1 was a tremendous sales success, even though *Consumer Reports* derided it for poor sensitivity and sound quality. Other manufacturers, astonished by the marketing success of the TR-1, quickly began making their own radios. Some makers — like Bulova and Mitchell — decided to market the TR-1 under their own names and, in some cases, new packaging (see Figure 8). These radios used an identical chassis to the TR-1, but all lacked the earphone jack. All of these "clones" are more rare than the original TR-1.

Beginning in 1955, a flood of American-made radios began to hit the market. Even Japan was getting on the bandwagon. Sony Corporation produced its first transistor radio — the TR-55 — in 1955, but it was not marketed in the US. The first Japanese radio to hit US shores was the Sony TR-63 in 1957. Japan, with its lower manufacturing

Figure 6. Later, more common TR-1 packaging.

Resources

The Portable Radio in American Life, by Michael Brian Schiffer. 1991, University of Arizona Press.

Collector's Guide to Transistor Radios (Second Edition), by Marty and Sue Bunis. 1996, Collector Books.

The Regency TR-1 Family, by Eric Wrobbel. 1994, privately printed.

Made in Japan: Transistor Radios of the 1950s and 1960s, by Handy, Erbe, Blackham, Antonier. 1993, Chronicle Books.

Transistor Radios 1954-1968, by Norman Smith. 1998, Schiffer Publishing, Ltd.

costs, quickly dominated the market. US radio manufacturers held on for a few years by moving production to Japan, but most had given the market up by the early 1970s.

Transistors Change Our Lives

The impact of the transistor on our everyday lives cannot be overstated. Their use in portable radios made broadcasts much more accessible and less costly, in addition to making them far more mobile. As a result, in ever-greater numbers, these devices were purchased for and by children — and retailers soon realized they had children as a major audience. Music evolved as younger listeners came to dominate radio audiences.

Figure 7*.* Clear backed TR-1. Note that the clear case has the dimple ground in it, even though the tuning capacitor no longer has the setscrew.

Soon, the transistor carried over into virtually every product that used tubes. Portable televisions soon appeared and, as time went on, increasingly sophisticated electronics using transistors became available to the average consumer. Audio gear, televisions, appliances, and —eventually — computers and cellular phones were mass marketed. Today, virtually no electronic device is built without transistors. These devices have become ever smaller, with the average dime-sized computer CPU chip containing millions of them!

Collecting Transistor Radios

Collecting early transistor radios is a fun hobby. TR-1 radios are not inexpensive (examples of the standard colors, in good condition, can sell for anywhere from $200.00 on up), but one can easily begin collecting radios that are priced from a dollar or so. So many different models have been made since 1954 that no reference even attempts to list them all. Virtually every collector chooses a field of specialization based upon his or her interests.

For example, many collectors focus on something called "Boys' Radios." These radios — which contain two transistors or less — were a Japanese invention to get around the high import tariffs charged on radios by the US. Any radio with less than three transistors was classified as a "toy" and so was taxed at a much lower rate. These "Boys' Radios" can have some very enchanting designs.

Figure 8*.* TR-1s in sheeps' clothing. From left to right: Bulova 250 in leather, Bulova 250 in white plastic, and Mitchell 1101 in suntan leather.

Several reference books on the subject are available; see the resources list. There are also many fine websites on the Internet devoted to radio collecting and they can supplement reference books because there are usually many photos of models. NV

About the Author

Sarah Lowrey has been an avid radio collector since childhood. She has devoted her energies to a website about her passion, transistor radios. Sarah's Transistor Radios is on the web at **www.transistor.org** She welcomes correspondence at slowrey@transistor.org

CLASSIFIEDS

CLASSIFIED ADVERTISING

$50.00 Per Inch — No extra charge for color *(Limited time offer).*
Classified ads must be paid in full prior to the closing date. Visa/MC/Amex accepted. Payment for ads received after the closing date will cause the ad to be placed in the following issue, at our discretion. Minimum charge is one inch with half-inch increments.

No proofs will be sent. Ads to be typeset by *Nuts & Volts* must be received by the closing date. Supplied ads must be received by the artwork due date.
Call the office at 951-371-8497 or Email classads@nutsvolts.com for closing dates, available sizes, and special prepay discount offers.

Components

Membrane Keyboards/Switches

Speakers

Printer Supplies

Miscellaneous Electronics For Sale

Amateur Radio

Computer Hardware

CLASSIFIEDS

Security

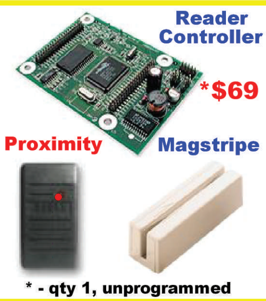
www.matco.com
Miniature CMOS Camera
CML-100/CMP-101
1/4" CMOS, 380 TVL,
3 Lux/f2.0, 3.6/5.5mm Lens
B/W: $29/ea Color: $39/ea
Color Mini Pinhole Camera
BX-123LC/PC
SONY 1/3" CCD,
380 TVL, 0.1 Lux/f2.0,
3.6 mm Lens $69/ea
Size: 0.9" X 0.9" X 0.5" (with Audio)
Super Hi-Res Color Lens Box Camera
BX-130LC-HR
SONY 1/3" CCD,
470 TVL,
0.4 Lux/f2.0,
$189/ea 3.6 mm Lens
(800)719-9605 sales@matco.com

Surveillance
Counter-spy Equipment
Security Cameras
Telephone Recorders
Online Catalog Only
www.MJelectronics.com
Tel: 914-699-2294

www.matco.com
Color Super HAD CCD Camera
SX-920-C-SH
$89/ea
1/3" Super HAD CCD Sensor; Digital video processor; 380 TV Lines; 0.2 Lux/f1.4; Supports Auto Iris lens for both DC drive & video drive; AES, AGC, AWB, BLC; Lens and bracket not included.
(800)719-9605 sales@matco.com

consumertronics.net
Hi-Tech Survival Offers!
Electronics, security, energy, computers, Net, phones, medical, legal, financial, weird. **Cat.$1**

Card-Access



* - qty 1, unprogrammed
www.kadtronix.com

Join our
ONLINE
electronics forums!
www.nutsvolts.com

Education

Command-Line Compilers start at $125
Windows IDE Compilers start at $350
Introducing the NEW MACH X Programmer
- Supports PIC12, PIC14, PIC16, PIC18 and dsPIC
- Reads HEX, COD, COF, and BIN files
- USB interface--No separate power supply required
- User selectable verify voltages (2V to 5.5V)
- Programs DIP chips with ZIF socket on unit and supports ICSP
- Start/Repeat push-button on unit for easy volume programming
- Can be used as an ICD debugger with the CCS Windows IDE Compiler

Only $149 with compiler and $199 without compiler

*Check out our new Hardware and Software Development Kits on the web at www.ccsinfo.com/hardware.shtml

CCS
262-797-0455
www.ccsinfo.com/picc

PIC® and PICmicro® are registered trademarks of Microchip Technologies Inc., in the USA and in other countries.

Audio/Video

www.matco.com
4 Channel Quad Digital Video Recorder
DVR-2200-M
$349
Recording rate: 30 fps full size/240 fps quad size; Up to 250 GB optional removable HDD; Wavelet compression format; Motion Detection recording; Alarm in/out; Video loss detection; Multi-quick search; POP/PIP; RS232, 485 protocol. *Quality Item @ Lowest Price*
(800)719-9605 sales@matco.com

www.matco.com
4 Channel PC Based Digital Video Capture Card
New Low Price
Plug & Play
Internet Accessable
DVRC-XP4
$85/ea
30 fps, 640 x 480, 4 RCA Video Input, 1 Audio Input
(800)719-9605 sales@matco.com

www.matco.com
Pocket Digital Video Recorder
Wired Remote Control
- Built-in HDD for 40hrs real-time recording in MP4
- Wired button & IR Remote Control
- A/V in/out, USB interface
- 3.5" TFT LCD monitor
- 5-1/8" x 3-3/8" x 13/8"
- Weight 0.67 Lb.

$599
(800)719-9605 sales@matco.com

Affordable Robotics Training Courses in:

Basic Electronics
Digital Electronics
Relay Control
Servo Controllers
PLC Systems
Hydraulic Systems

From Basic to Advanced!

WWW.UCANDO-CORP.COM

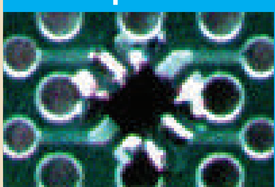
1-800-678-6113

FREE SHIPPING!

UCANDO VCR Educational Products Co.
(Est. 1988)

Design/Engineering Services

OK, so now we have the world's smallest Microprocessor in SOT 23-6 pin, smallest Op Amp in SOT 23-5 pin, smallest Power FET in SOT 23-4 pin. But, how do you work with them without spending an arm and a leg??? ONE PASSircuit™ !



www.onepasinc.com

*Circuit board layouts
*Prototype assemblies
WWW.OSPREYELECTRONICS.COM
Convert your sketch or print into a quality pcb for a reasonable price. Visit us on the web or call Osprey Electronics at (208) 665 1688 (USA)

Electronic Engineering Consulting
Product development from concept to completed circuit board or product. Digital and analog expertise. Microcontroller design for Motorola, Intel, TI ultra low power Microcontrollers, In-House schematic capture and PCB layouts.
Dave Lancaster
Infotech Controls
P.O. Box 2822
Lower Burrell, PA 15068
or call (724) 337-8865
Infotechcontrols@comcast.net

SERVO
The Future of Robotics is at Your Fingertips!
Subscribe Today!
www.servomagazine.com

Call us TODAY to start your subscription to *SERVO Magazine!*
800-783-4624

Plans/Kits/Schematics

Five Volt DC 100mA Kit
Measures 1-3/4 x 1-5/8.
Kit contains bridge rectifier, all filter caps, and LM78L05 regulator.
$9.95 ea. postpaid.
G. Joseph Finn, Jr.
244 Mistwood Ln.
N. Aurora, IL 60542

CCD/Cameras/Video

USB 2.0 Camera module with SDK
Mega pixel high-speed USB2.0 camera module. Capture size from 320x240 to 1280x1024 at real time speed. Complete SDK, highly customizable and flexible. On board hardware resources. Price from $69 to $159. Suitable for all kinds of computerized image capturing project.

For more information, visit our website
www.fclab.com

Vacuum Tubes

WANTED
To buy **COMPLETE** *vacuum tube collections, **ANY** quantity. Will buy new, new w/o boxes, and used. We buy all types and will not cherry pick your collection. Will travel to inspect and pick up large hoards.
Paul, Sound Ideas, 3215 NW 13th Street, Gainesville, FL 32609.
pwb@soundideasstereo.com
*please list tubes in the note line of your email
352-378-0192
FAX #1 352-371-1791
FAX #2 352-336-6821
(10 am-7 pm EST M-F, 10 am-5 pm Sat)

(continued from page 20)

— to date — it has been limited by the cost of the technology. I look forward to the day that I can dial my cell phone by pressing on the sleeve of my shirt.

Out of the Frying Pan ...

And into Firefox, the latest web browser release from the developers at Mozilla. The history of browsers has been an interesting one and the recent release of Firefox undoubtedly begins a whole new chapter in the subject.

Dogged by constant security problems and an unending stream of patches, Microsoft's Internet Explorer (IE) is quickly losing ground to the more secure Firefox. Like IE, Firefox is a free download, but it includes a gaggle of cool features: integrated pop-up blocking (my favorite), tabbed browsing, Google search built right into the toolbar, and — most importantly — improved privacy and security by not loading harmful ActiveX controls. With the constant arms race between spyware and firewall authors, you can't have too many layers of protection on your computer.

Firefox is available for Windows, Linux, and Mac OS X at **www.mozilla.org**

CANopen in the Open

Have you ever heard of Controller Area Network (CAN)? It was developed over a decade ago by the Robert Bosch, GmbH. (German for "Inc."), to reliably move data in automobiles and was eventually adopted as an international standard — ISO 11898.

Currently, it is used throughout various industries as a high speed serial bus system, which is particularly well suited for networking "intelligent" devices, as well as less-intelligent sensors and actuators within a system.

If you are interested in using CAN in your next automation or robotics project, check out IXAAT Automation's new website, **www.canopensolutions.com** Their web page covers CANopen basics — like protocol structure, communication mechanisms, and network management. It also includes free articles, downloads, and information on training and events for those who really want to dive into CAN.

Tech Forum

QUESTIONS

I picked up a Zenith TV — model H2017Y — from a hotel that was remodeling. I can't get the AV inputs to work; it looks like some programming or setup is required. I have no owners' manual and would appreciate any help.

#10041 **Oscar Loya**
via Internet

When the BePC came out about 10 years ago (remember those?), there were two vertical columns of LEDs on its front panel. One lit to show CPU activity level; the other showed (I believe) memory activity. It was quite a show! I'd like to make something like that for my PC and I wonder what would drive the meters?

I seem to remember an Internet project a few years back for adding a "tachometer" to a PC, via an automotive needle/meter tach, so I know it must be possible to do it.

#10042 **Phil Combs**
via Internet

I have a vacuum pump that uses four D cells in series for a total of 6 VDC. I want to build or acquire a battery eliminator for the pump. How do I determine the amps required for the pump? The unit has a PCB with a NEC D882P, an IC that has had the markings sanded off, and various passives.

#10043 **Bill Wagner**
Silver Spring, MD

I have a 35" Sony TV with PIP. Is there any way to add (build or buy) progressive scan to the TV so it can use the enhanced output from a DVD player?

#10044 **Leon Lombrozo**
Sunnyvale, CA

I got a new computer with WindowsXP on it. (My old computer had Windows 98.) I use a DOS program to communicate with a PLC (programmable logic controller) through the serial port. The DOS program loads and works, but it will not communicate through the serial port. Different settings have been tried, in addition to Windows 98 compatibility mode. The program tries to communicate with the PLC, but it times out. If I unplug the serial cable, it gives a communication error.

How do you get the XP serial port to work using older programs? Is there a way to get to a C:\ prompt before XP loads, like in 98? Can the hard drive be partitioned in XP and use 98 for older programs?

The manufacturer of the software does not have an answer, but they want to sell me new software that will run under XP — though they removed the features I use the most.

#10045 **Keith Berning**
via Internet

I need help on a phone problem. I just installed new AT&T 2.4 GHz phones: a base and three handsets. Now, if all the phones are in their charging cradles, I get big double horizontal bars on TV channels 4 and 5 and buzz on my portable FM radio. If the phones are out of their cradles, everything is okay. The manufacturer could not offer any advice, but I hope someone here can!

#10046 **Don Barbour**
Kentfield, CA

I've just picked up a couple of 6" neon light tubes that I want to put into my PC. The tubes were sold for car use and have power converters — 12 VDC/150 mA in, 1,000 V/15 mA out. I could buy a module that will "blink" the lights to music, but I want a different effect.

I want the lights to appear to "breathe" — slowly dimming to some

---

This is a READER-TO-READER Column. All questions AND answers will be provided by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving problems of a technical nature. All questions submitted are subject to editing and will be published on a space available basis if deemed suitable to the publisher. All answers are submitted by readers and NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals. Always use common sense and good judgement!

Send all material to *Nuts & Volts Magazine*, 430 Princeland Court, Corona, CA 92879, OR fax to (951) 371-3052, OR email to forum@nutsvolts.com

ANSWER INFO

• Include the question number that appears directly below the question you are responding to.
• Payment of $25.00 will be sent if your answer is printed. Be sure to include your mailing address if responding by email or we can not send payment.
• Your name, city, and state, will be printed in the magazine, unless you notify us otherwise. If you want your email address printed also, indicate to that effect.
• Comments regarding answers printed in this column may be printed in the Reader Feedback section if space allows.

QUESTION INFO

To be considered

All questions should relate to one or more of the following:
1) Circuit Design
2) Electronic Theory
3) Problem Solving
4) Other Similar Topics

Information/Restrictions

• No questions will be accepted that offer equipment for sale or equipment wanted to buy.
• Selected questions will be printed one time on a space available basis.
• Questions may be subject to editing.

Helpful Hints

• Be brief but include all pertinent information. If no one knows what you're asking, you won't get any response (and we probably won't print it either).
• Write legibly (or type). If we can't read it, we'll throw it away.
• Include your Name, Address, Phone Number, and Email. Only your name, city, and state will be published with the question, but we may need to contact you.

adjustable level (maybe 30-40%) and then going back to full brightness again without pause. What would I need to accomplish this effect?

#10047 **Phil Combs**
via Internet

I need a converter to change analog component video to S-Video and composite video. I haven't found a commercial unit to do this, so I want to make my own. There is an IC from Philips Semiconductor — the TDA8501 — which looks like it would do, but I can't find a place that will sell a small quantity. Does anyone know of a source?

#10048 **Anonymous**
via Internet

ANSWERS

[4046 - April 2004]

I need a timer to activate a remote feeder once or twice a day for a one minute period for several days. No AC power is available.

Intermatic makes an Electronic Programmable Timer that runs on a single AA cell and has "hard" switch contacts that can be used on 120-277 VAC, 28 VDC, or 12 VDC at 4 A. It has six ON/OFF cycles that can be set for every day or spread over the days of the week. Go to **www.intermatic.com** and do a search for EI20C. Sales data and an instruction sheet are available.

Denis Kuwahara
Port Orchard, WA

[5042 - May 2004]

I'd like a schematic of how to connect a transmitter/receiver made by ***www.radiotronix.com*** *and sold by Mouser Electronics to use as a USB (v.1.1) wireless transmitter/receiver for my printer and/or scanner.*

There are several problems here. USB devices are packet based and configuration and setup data are sent in both directions. The PC needs to send the printer the data that is to be printed, but the printer also needs to be able to send back various USB device ID and status messages, such as information about the printer's make and model number and messages for "Out of Paper" or "Paper Jam." So a transmitter on one side and a receiver on the other side won't be enough, since the communication needs to be bi-directional.

Another problem will be that USB devices (the Printer/Scanner) expect to be connected to a USB host Controller (or a USB hub). The host controller is responsible for keeping track of and controlling the various states a device can be put into — powered, configured, suspended, and others. For more information on this, a copy of the USB specification can be obtained from: **www.usb.org/developers/docs/**

Phillip Stevens
Pocasset, MA

[7043 - July 2004]

I work in electronics repair. Sometimes, we do not have documentation for the equipment and we have a problem identifying SMT devices (like the SOT-23 style) that have only a device code — like "R2C." Is there some universal standard for these codes and, if so, where can I find them?

#1 I have at least a partial answer to this question. I have five Excel files detailing cross-reference numbers for SOT transistors. Manufacturers included are Siemens, Phillips, NEC, Motorola, and National Semiconductor. The files comprise about 1.3 MB and I have placed them on my website, **www.bolingeng.com/nuts__volts.htm** for *Nuts & Volts* readers.

Harry Boling
Boling Engineering Associates
Garland, TX

#2 These three to six legged SMT pigs were hatched by Phillips Semiconductor back in 1971 and can contain anything from one or more diodes, bipolar transistors, or FETs to digital gates and op-amps! As reverse engineers, we have done a lot of data book research on them. We have several thousand of the one to four digit stamping codes for these.

Unfortunately, they are on index cards, rather than a reproducible list. We have seen several lists floating around, but they are skimpy and usually confined to only a few manufacturers. Readers can Email us (**rolo@trib.com**) and we will try to tell you what the device is and provide a manufacturer's number, if possible.

Incidentally, the R2C you used as an example shows up as PNP transistor SSTA70 made by Rohm,

MMSTA70 by Motorola, or a 2.5 volt, 0.5% voltage reference LM4040C1M3-2.5 made by Micrel!

Bomarc Services
Casper, WY

#3 The SMD Codebook (**www.marsport.demon.co.uk/smd/smdcode.htm**) is a reference for surface mount device semiconductor device codes, equivalents, and connections. To identify a particular SMD device, first identify the package style and note the ID code printed on the device. Now, look up the code in the alphanumeric listing that forms the main part of this book by clicking on the first character shown in the left-hand frame on the website. A scrollable page of data will appear in the main frame.

Unfortunately, each device code is not necessarily unique. For example, a device coded 1A might be either a BC846A or an FMMT3904. Even the same manufacturer may use the same code for different devices! If there is more than one entry, use the package style to differentiate between devices with the same ID code.

Massimo Sernesi
Grosseto, Italy

[7044 - July 2004]

What is the easiest way to allow my computer to operate 120 VAC motors and lights?

#1 The simplest way to use a computer to operate 110 volt devices is through the use of X-10 devices. These are available from many sources and one with the best assortment is Smarthome (**www.smarthome.com**). RadioShack also has some units.

There are a variety of interface units and software that operate through either a serial or USB port. The signals are sent over the house wiring so that the items to be controlled need not be near the computer.

Some interface units allow programmed sequences of controls to be stored and eliminate the need for the computer to be on for the sequences to run; you can set up a sequence of lights that go on and off while you are away or you can control your sprinklers. Use is only limited by your imagination.

Jim Schmidt
Deer Lodge, MT

#2 A partial solution to your problem was described by Ryan Sheldon (**www.nationalcontroldevices.com**) in the February 2002 issue of *Nuts & Volts Magazine*. His system is based on a small chip that connects to your computer's serial port and a pair of servo motors. That will allow control of tilt and pan on one camera. According to the article, the system is expandable to 256 devices from one serial port. Depending on your camera, it may be possible to use an additional chip and

another pair of servos to control focus and iris setting.

Dave Sarraf
Elizabethtown, PA

[7047 - July 2004]

Can anyone suggest a good transistor driver or other circuit that could PWM control halogen lamps (up to 12 amps) at 13.8 V? I will be controlling this driver with a pin on a Parallax BS-2.

You just need a logic level MOSFET, STP40NF03L, available from Mouser Electronics (**www.mouser.com**). Efficiency will be maximized if you run at low frequency — just high enough that you don't see flicker (60 to 100 Hz).

Russell Kincaid
via Internet

[8045- August 2004]

I built a science project called a "NASA Power Factor Motor Controller." It was designed to save about 60% on our electricity bill and was based on a project published in the October 1979 issue of Popular Electronics. I followed every detail of the construction article and — even after troubleshooting it for weeks — it fails to work. Review of the design by an engineering professor revealed that the design doesn't practically function, even though NASA owns the patent (4,052,648)! He felt that this project was just another "perpetual motion" boondoggle. If anyone has built this project and was able to make it work, please explain how?

#1 It seems that the concept of power factor is poorly understood. If you have a 1.0 power factor, then you have a purely resistive load and the rarely achieved ideal in the real world. Most loads — such as motors, rectifier circuits, and transformers — are sometimes highly non-linear, causing currents to flow back and forth within one cycle, loading up the generator and distribution grid, but not doing any real work. Therefore, utility companies get less out of their generators and power lines.

In industrial settings, it is common to correct for a poor power factor with switched capacitor banks to reduce the utility bills. In consumer settings, there are power factor correction control ICs — such as the Unitrode UC3854A and UC3854B — to correct the power factor to nearly one for switched mode power supplies (SMPS), such as used in your PC, TV, or printer.

SMPSs with power factor correction are slowly appearing and are just a few bucks more expensive than their non-corrected counterparts. They are also finding their way into newer appliances. Correcting for a poor power factor in a home setting with an old circuit is a daunting task. You can find further information about power factor correction at the Unitrode, International Rectifier, and Texas Instruments websites. Take a look at a spec sheet for the relationship between load and power factor for a motor. You should be able to find one in the literature or at a motor manufacturer's website.

Walter Heissenberger
Hancock, NH

#2 I, too, was interested in the power factor controller concept back in 1979, when I was employed as the Chief of a NASA testing laboratory for NASA. I am an electrical engineer by degree, but — instead of building one myself from the patent information I received from a NASA Tech Brief — I ordered two from:

Electronic Relays, Inc.
1438 Brook Drive
Downers Grove, Ill 60515

I installed the units on my AC and they operated successfully for several years. The AC units had 1/2 HP motors, which was the maximum rating of the power factor controller.

I discovered that I still have one of the units in my electronic junk box, so I can attest to the fact that the power factor controller circuit worked as advertised and did save on my power bill, but I can't remember how much. I don't think it was 60% — more like 40%.

It most definitely was not a "perpetual motion" boondoggle, as you suspect. I would guess that you can still get a copy of the NASA Tech Brief from NASA.

David L. Pippen
Las Cruces, NM

The Business of Electronics Through Practical Design and Lessons Learned

In The Trenches

...You Might Be an Engineer

The characteristics that are associated with engineers are both stereotypical and somewhat accurate. This month, we'll take a not-too-serious look at some of these traits to see what they are. After all, it's important to understand how others see the profession, as well as what features make someone a good engineer. (Note that the pronoun "he" is used for simplicity and brevity. Most hardware engineers are male; however, more women are entering the field every day.)

The Archetypical Engineer

I was at a high school career day recently and the question was asked, "What type of person makes the best engineer?" That stuck with me for some time. An engineer is more than someone who likes math and science. There does seem to be an engineering personality. Of course, that's not really too surprising. It's natural for certain types of people to be drawn to certain types of jobs.

So, if you go into the basement only to find that half of it is submerged in deep water and your first thought is, "I didn't know the floor was so tilted." You might just be an engineer. (My apologies to Jeff Foxworthy.) Engineers notice things others don't and they apply physical principles subconsciously. Others just see the water, the ruined possessions, and the difficult clean-up.

There are few realistic role model engineers. Popular versions range from The Professor on *Gilligan's Island* to Scotty on *Star Trek* to Samantha Carter on *Stargate*. These characters can create a transmitter from coconuts or time machines, respectively.

Yet, they are all incapable of leaping a tall building in a single bound. (Which, actually, seems much easier to do.) Remember, this flattering image is what many people truly think engineers really are. These people don't understand about specialization or learning curves.

Mathematics

If you buy some items that cost a total of $12.87 and give the salesperson $18.12 to simplify your change ... you might be an engineer. Math is easy for engineers. It's second nature. Admittedly, some would say it's first nature.

Engineers are always using math — mostly simple arithmetic — but continuously. An engineer can perform many calculations mentally. In-your-head conversions of Fahrenheit to Centigrade or millimeters to inches and frequency to wavelength are common.

Engineers have a feel for numbers. They know what a reasonable value is and what it isn't. Often, they can just look at a column of numbers and determine if the sum is accurate. Of course, no real engineer would be far from his calculator. It's a vital part of his anatomy. Taking away an engineer's calculator is defined as torture under the Geneva Convention. They can't sleep and they lose their appetites.

Then, they spend their time generating logarithm tables by hand

and trying to remember the sine values for various angles. So, if you know every function and feature of your scientific calculator ... you might be an engineer.

A complicated calculator is not the only toy an engineer carries. Engineers like high-tech gadgets ... pager, MP3 player, laser pointer, web-cam, USB drive, GPS locator, etc. The Dilbert cartoon about the "belt-appliance" competition is not too far off the mark. If you avoid deep water because: A) All the belt hardware will drag you to the bottom and you'll drown, B) the water will ruin all your neat toys, and C) you have difficulty deciding whether A or B is worse ... you might be an engineer.

Learning

Engineers like learning new things. Unlike many people who turn off their brains after they finish schooling, engineers continue their education — both formally and informally. They have to. The half-life of technical expertise is usually considered to be five years. That is, half of any technical subject will be obsolete in five years. Obviously, this means that any engineer must constantly refresh and update his education or else he'll be obsolete. This is a major reason why older engineers have trouble finding jobs. Many employers assume — erroneously — that the older engineer hasn't kept up. So, instead of actually interviewing that person to determine the truth, they simply dismiss the candidate.

The good engineer has an ever-expanding bookcase of technical manuals, data books, application notes, and conference proceedings. Of course, with the proliferation of CD data books and the Internet, the engineer's library may no longer look like one of a few years ago. Nevertheless, if you read a textbook for enjoyment ... you might be an engineer.

Engineers are meticulous and truthful. They have to be. A product that fails is never a good thing. Sometimes — like O-rings and rocket boosters — a failure can be catastrophic. Unlike many other jobs, an engineer cannot shift the blame for a failure. Either his design works or it doesn't.

The responsibility ultimately rests with the designer. He can't say he didn't know, that it wasn't his responsibility, or that he wasn't informed. An engineer's design is truly his brainchild. He has spent a lot of time developing the design and is proud when it works properly.

This is why engineers often have a difficult time with marketing, romance, and other social interactions. The honest and whole truth is not always appreciated by other non-engineers. If your girlfriend asks, "Do these

pants make my fanny look big?" and you answer, "It looks just as big as always" ... you might be an engineer.

Marketing and engineering have a special relationship ... a bad, but necessary one. Engineers always want to know the limits of a product and assume that everyone else wants that information, as well. Marketing wants to sell the product, regardless of its properties or performance. At a sales meeting, an engineer might actually describe — in detail — every fault with his own product while identifying the strengths of the competitor's. This honesty drives marketing crazy.

On the other hand, marketing ignores the product's weak points, emphasizes the strong points, and often makes claims that can't be supported. This causes the engineer to see marketing as something not much better than a pimp.

Shopping

The only things an engineer really shops for are technical products. Everything else is just a chore to be completed as quickly and efficiently as possible. Food should be palatable, cheap, and easy to fix and clean up. Clothing should fit well enough not to be a distraction. Colors, patterns, and styles are usually bright, bold, and out-of-date.

Shopping for high tech items is very different. If you ask a sales person a question that you already know the answer to just to determine that person's level of expertise ... you might be an engineer. There's often a gentle battle of egos that ordinary people are oblivious to. The shopper and sales person exchange questions and comments that comprise a special language. For example:

Shopper: "Is the serial port RS-232 or USB?" [How much do you know?]

Salesperson: "USB is standard; Firewire is an option." [Stop being an idiot; no one uses RS-232 any more.]

Shopper: "I do work at home." [Is it suitable for simulating nuclear devices?]

Salesperson: "It's got a 56K baud modem and uses a 300 MHz processor." [Don't even think about it. It's a dog.]

Shopper: "What else do you have?" [Show me your best deal.]

Salesperson: "This is inexpensive, but it's not expandable." [Great buy! The manufacturer put a seal on the case to stop morons from screwing up the system, but it's got a standard motherboard that has lots of free slots.]

Engineers are preternaturally curious and handy with tools. If you like the statement "some assembly required" or if you immediately voided the warranty on some product because you took it apart as soon as you got home ... you might be an engineer. Many engineers would rather understand how something works than have that something actually work.

Taking apart a faulty laser printer is a win-win situation. You get to see all the neat optical and electrical parts, as well as the high voltage section. High voltage always holds a special place in an engineer's heart. If you fix it, that's great. If you don't, that's great, too. You got to spend an afternoon playing with a new toy.

Improving Things

Engineers can't leave anything alone. Everything can be improved or at least changed. The walls and ceilings should be painted with fluorescent material and the lamps made to emit ultraviolet so that everything would glow, creating a very even light. A pencil should be made entirely from graphite; then, it would last a lot longer and wouldn't have to be sharpened as much. Just rub a spot until it becomes pointed enough. Words should be spelled completely phonetically. That would simplify learning and reduce spelling errors. Obviously, these ideas are not practical. They are just an exercise in creativity. Engineers like talking about absurd ideas in a way that appears to be serious. This can frighten outsiders who are not aware

that this is meant as banter.

Since nothing is ever perfect, a project can always be improved. After all, once you build the first prototype, you have all this new experience that you can apply toward a second prototype, then the third prototype, and so forth and so on. This results in a never ending series of prototypes and no finished product.

Managers hate this. To them, the product seems fine. It works according to specifications, is cost-effective, and should be going into production.

So, if you always want another couple of weeks to optimize the software or simplify the driver electronics or re-package it for easier production or standardize the design or reduce noise or ... you might be an engineer.

Conversation

Small talk is always difficult for engineers. They simply don't get the point of it. Real talking is information exchange; social conversation about topics like war, politics, world hunger, current fashions, sports celebrities, music, and TV shows (except for *Star Trek*, *Stargate*, *Junkyard Wars*, and *Monster Garage*) holds no interest because an engineer has no control over them and because opinions are not real information. (Unless the engineer works for a defense contractor. Then the topic of war has some meaning.)

Engineers also have difficulty talking about the three topics that typical men hold dear. Their jobs are the first such topic. Must men will wax eloquently about their jobs and make it appear that they are saving the world when all they really do is scrape gum off the grocery store floor (not that there's anything wrong with that).

Of course, many engineers do have jobs that are saving the world. Unfortunately, because of security and non-disclosure agreements, they can't discuss their anti-missile design or their product that detects cancer.

The number two topic for men is sports. Not sports they actually play — sports they watch on TV. Engineers like to do things. Talking about someone else doing something seems like a waste of time.

The third topic is sex. Naturally, this starts out as fiction and rapidly progresses to fantasy. Engineers are too honest to make up stories and the real thing is much too personal to share with casual acquaintances. So, if you're at a party and feel alone and left out and wonder why everyone seems to be having a good time ... you might be an engineer.

Humor

The idea that engineers actually

have a sense of humor is mind-boggling to many, but engineers do have a great sense of humor. It's just that what they think is funny isn't funny to others. Engineer humor is either extremely physical or very arcane. An engineer would love hacking into his boss's telephone service to change the answering machine message to, "I'm a lumberjack and I'm okay!"

Alternatively, putting an extra key on his computer keyboard labeled "ANY" would be hilarious. The more complicated and unique the joke, the better.

Here's a (mostly) true story. The names have been changed to protect the guilty. Joe was getting ready to apply power to a very expensive and complicated design for the first time. Unknown to him, Bob had come in the night before and surreptitiously run a 1/4" plastic tube from inside the project, down the back of the desk, through a wall, and into a second room 20 feet away. From this vantage, Bob could watch as Joe applied power.

At just that time, Bob exhaled a large lung-full of cigarette smoke into the tube so that it billowed from the bowels of the machine just as it was turned on. Uproariously funny! It has now become a company tradition that everyone has to exhale before a new design can be powered-up for the first time. If that's amusing ... you might be an engineer.

The Opposite Sex

First, while the number of women in hardware engineering is growing, it is a relatively recent phenomenon and there is little data on which to perform a proper analysis. It does seem that female engineers tend to marry male engineers. (The result of this inbreeding is not yet known.) Therefore, this discussion will focus on male engineers.

There is some truth to the rumor that engineers start their lives as nerds and geeks. Their social life in high school and college is often rather meager. All the girls go after the football or baseball players. Unfortunately, these ladies realize too late that their first choice is often a bad choice. Very few of these athletic types become professional athletes and even fewer succeed at that.

The result is that, in a few years, the jocks' muscles go to fat and their outlooks on life darkens. They end up getting jobs as used car salesmen and spend the rest of their lives watching sports on TV and lamenting at how they missed their opportunities. Conversely, engineers become more attractive as they get older. There are a number of reasons for this. Probably the most important is that self confidence grows with time. Being able to build a new bridge, aircraft, computer, or medical instrument does wonders for your ego. Also, confidence is very attractive.

Another factor is that, for some reason, hardware engineers tend to be slim. (Software engineers seem to be rounder.) Since they start out slim, they generally stay that way. So, while the ex-athletes fill up their bellies with beer, engineers have snacks of Ding Dongs and Twinkies instead of meals. There is even some evidence to suggest that all those junk food preservatives keep engineers young and fresh, too.

Engineers also know the value of exercise because they're smart. A surprising number belong to gyms and work out on a regular basis. Racquetball, jogging, bicycling, and other non-team sports are preferred. Clearly, being in good shape is very attractive.

Women are always attracted by wealth and power. (Rolling up $100.00 dollar bills and sticking them into your ears will attract a surprising number of women.) An engineer's salary is pretty good. Many engineers are promoted to management positions and it's common for a company VP to be an engineer.

Lots of engineers create their own successful companies. Current examples are Microsoft, Apple Computer, Hewlett/Packard-Agilent, and Xilinx. In fact, most technically-based corporations were started with and by engineers.

It takes a few years for women to realize this and for the engineer to build a few steps in his career. That's why engineers become more attractive in their late 20s, up to their 50s, and beyond — just look at Bill

Gates. If you made a million dollars or owned a house or were granted a patent before you kissed a girl ... you might be an engineer.

Women eventually realize that engineers are really very desirable life partners. Engineers are good providers, stable, reliable, organized, and able to fix things around the house. Few engineers spend their free time and money at the bar with "the boys." Nor do they watch sports all weekend. This means that they share more time with the family and are good fathers.

Women see the negative traits of engineers as easily fixed. Once married, the wife will train the husband in how to dress properly, eat better, be romantic, and polish off some of the social rust. However, she will only make him socially acceptable so he will not embarrass her in public. She will *not* teach him to be socially adept. If she did, other women might find him too interesting. Lastly, women like to change men. It appears to be an inborn trait that gives them great satisfaction.

Score Yourself

Naturally, no engineer would accept any of this without some supporting evidence. Score one point for every "Yes" answer and zero points for a "No" to the following questions:

1. As a kid, you bought the X-ray glasses advertised in the back of the comic book in order to see how the TV worked.

2. You know the products at RadioShack better than the salesperson.

3. No one at a party thinks your boss's attempt to use a magnet to pick up a stainless steel screw is funny except you.

4. You actually read the VCR instruction manual.

5. You have more than one computer.

6. You know every *Star Trek* movie and TV episode (including all the spin-offs) nearly verbatim.

7. You kept your college textbooks.

8. You are annoyed by contradictions in science fiction movies. (Like when someone effortlessly passes through walls, but doesn't sink through the floor.)

9. Your job is also your hobby.

10. You think paisley is a color.

If you scored seven or more ... you might be an engineer. **NV**

The Latest in Networking and Wireless Technologies

Open Communication

Short Range Wireless Options

In my August column, I wrote about the new ZigBee wireless system. In this issue, I want to complete the coverage of the various short range wireless options available today.

Currently, there are no end products using ZigBee, but you can expect to see some late this year and early next year. Bluetooth is also an option, as is ultra wideband (UWB). I have written about both of these in previous issues; however, there are other short range wireless products that are widely used. These include the UHF radios, WirelessUSB, and the ever popular 802.11b. That's what I will focus on here.

UHF Radio Modems

The license-free radio bands as defined by the FCC's (Federal Communications Commission) industrial-scientific-medical (ISM) frequencies are used for all sorts of wireless applications. The most common frequencies are 315 MHz, 433.92 MHz, and the 902-928 MHz band, usually 915 MHz. The microwave band between 2.4 and 2.483 GHz is also ISM designated. These are defined in Part 15 of the FCC's rules and regulations.

One of the earliest uses of the UHF radios was as garage door openers. Most of them still use 315 MHz. Another common application is remote keyless entry (RKE) on automobiles and trucks. This feature has become a standard option on nearly all cars. Tens of millions are in use.

Another use for them in autos is in tire pressure sensing. High end cars — like the Corvette — come with a wireless transmitter in each wheel that transmits the air pressure to a receiver in the dash that displays the pressure in each tire. The actual pressure in the newer tires is critical for safety and long life.

Almost anything can be made wireless with these simple radios. A common use is as a wireless thermometer.

Also, in some high end consumer electronic systems, the old infrared (IR) systems — with their limited range and line of sight (LOS) requirement — are being replaced by more powerful radio frequency (RF) remotes using these circuits.

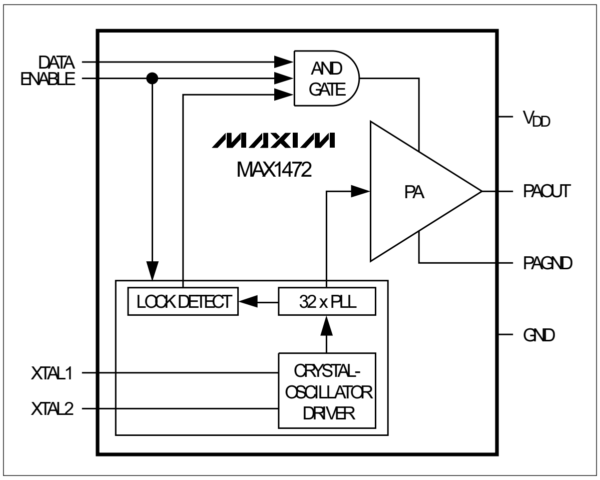
Typical of the wireless chips available are the Maxim Integrated Products MAX1472 transmitter and the MAX1473 receiver. The transmitter (Figure 1) uses an external crystal to set the frequency. For operation at the 433.92 MHz, a 13.56 MHz crystal is needed. The crystal is used as the reference for an internal phase-locked loop (PLL) that frequency multiplies the crystal by 32 to produce the 433.92 MHz output. Low cost crystals in the 9 to 15 MHz range can be used to achieve any frequency from 288 to 480 MHz.

The output of the PLL is then amplitude modulated in a class C amplifier. Most wireless chips are used to transmit binary data so the actual modulation is more accurately called amplitude shift keying (ASK). ASK is also called on-off keying (OOK). This is the process of just turning the carrier signal on for a binary 1 and off for a binary 0. That is the same as 100% modulated ASK. The data rate is usually pretty low — typically less than 100 Kbps.

The MAX1472 is designed to operate from a lithium cell in 2.1 to 3.6 volt range. With maximum voltage, the transmitter will deliver up to 10 dBm (10 mW) into a 50 ohm antenna load. The antenna is usually an inductor loop on the printed circuit board that resonates at the operating frequency.

In a simple wireless system using a chip like this, only a limited amount of information is transmitted. For example, in a garage door opener, a short — usually eight-bit — code is used. This is to ensure that someone else driving by cannot open your garage door with his or her transmitter. With eight bits, you can have $2^8 = 256$ unique codes, so the likelihood of your code duplicating a neighbor's is pretty low. A special encoder chip or a microcontroller is used to do

Figure 1. The Maxim MAX1472 transmitter. It is housed in a 3 x 3 mm, eight-pin surface mount SOT23 package that is a bear to solder to a PCB.



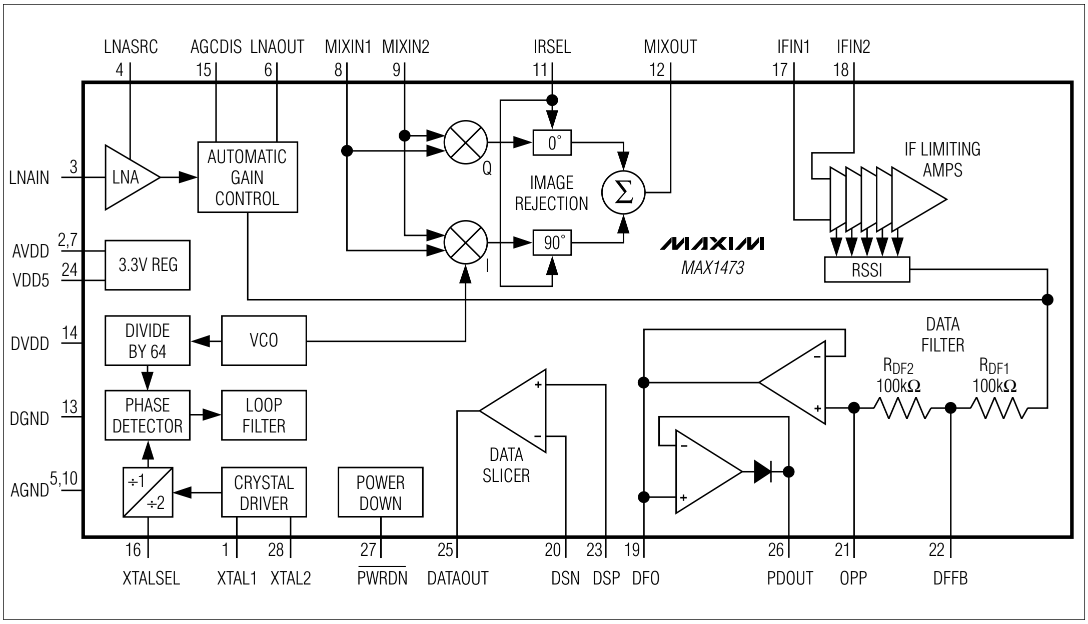


Figure 2. The Maxim MAX1473 superheterodyne receiver. The package is a 28-pin SSOP. The divide by 32 or 64 block represents the PLL local oscillator that takes an external crystal to set the receive frequency.

the encoding.

In some applications, each transmitter may be assigned an address — a special code that identifies the sender to the receiver. Again, eight bits is a common length, but other size address words are used. If you recall, ZigBee uses a 64-bit address word, allowing jillions of nodes to be identified.

The receiver recognizes the code and produces the desired result, either opening a garage door or your car door. In some applications, an identifying address is accompanied by some actual data. In a wireless thermometer, the data is a serial binary word representing the actual temperature value as derived from a sensor and an analog-to-digital converter (ADC).

In even more sophisticated applications, the address and data may be accompanied by some kind of error detection and correction scheme. Simple systems do not normally use this.

As for a receiver compatible with the MAX1472, there is the MAX1473. It is a superheterodyne type (Figure 2). The signal picked up by the antenna is boosted in strength by a low noise amplifier (LNA), then downconverted to an intermediate frequency (IF) of 10.7 MHz. The downconverter is a pair of balanced mixers connected to form an image rejection mixer. The mixers are driven 90 degrees out of phase with one another by the local oscillator and their outputs are added, causing signals at the image frequency to be cancelled.

Remember that an image is an unwanted signal that appears at a frequency that is two times the IF, higher or lower than the desired signal, depending upon whether the local oscillator frequency is above or below the received signal. Images are a real problem in almost any superheterodyne. The usual way to minimize them is to use a selective band pass filter at the antenna, typically a surface acoustic wave (SAW) filter. SAW filters are expensive and an unreasonable addition to a low cost receiver like this. The image reject mixer does the job instead. An inexpensive 10.7 MHz ceramic filter external to the chip provides the IF selectivity at the mixer output.

The local oscillator driving the mixers is a phase-locked loop (PLL) that multiplies an external crystal frequency by 32 or 64 to get the necessary local oscillator frequency that is 10.7 MHz above the desired

Figure 3. The Micrel MICRF505 transceiver that operates with FM in the 850 to 950 MHz ISM bands. Its tiny 5 x 5 mm package means you can build it into almost anything.

receive frequency.

The output of the IF filter drives a chain of IF limiting amplifiers to provide gain. A received signal strength indicator (RSSI) circuit provides feedback to operate an automatic gain control (AGC) feature to reduce the LNA gain for strong nearby signals. The demodulated signal is then fed to a low pass filter and data slicer to reproduce the serial binary data originally transmitted. In most products, this serial binary data is sent to an embedded microcontroller, where the remainder of the application is implemented. For simple on-off control type apps, no microcontroller is needed.

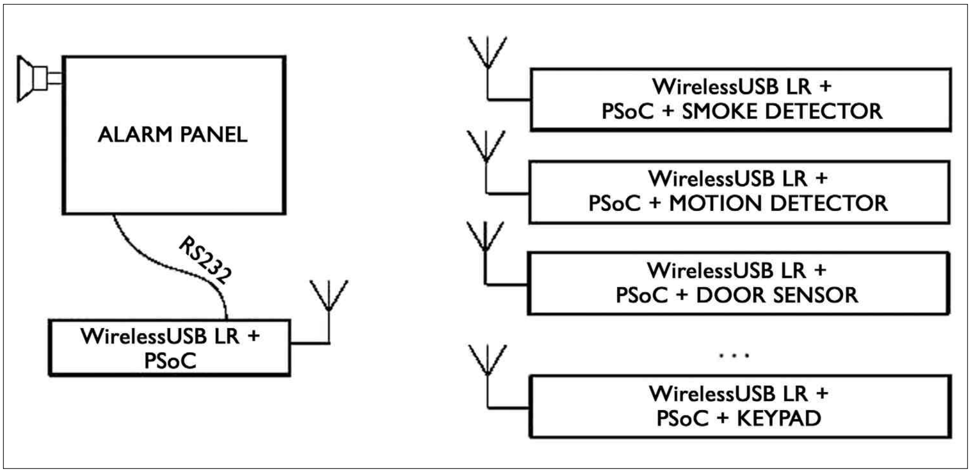


Figure 4. A typical industrial (or home) wireless application using Cypress Semiconductor's Long Range WirelessUSB chip. Multiple sensors send data to a central location for control purposes.

Maxim has many other wireless chips and you can get an overview by going to their prodigious website at **www.maxim-ic.com**

Micrel Semiconductor is another company making UHF radio chips. They make the 315 and 433.92 MHz radios, as well as a new transceiver for the 850-950 MHz range. It is designed to work in ISM Part 15 applications in the US 902-928 MHz band and the European 868 MHz band. Designated the MICRF505, this device is a full transceiver to be used in half-duplex bidirectional links (Figure 3). The transmitter is a PLL synthesizer that feeds a power amplifier.

The PLL synthesizer uses an external crystal and is fully programmable to any frequency in the 850-950 MHz range. The PLL is also very fast, making it useable in a frequency hopping spread spectrum (FHSS) application. Modulation is FSK with a data rate up to 200 Kbps. The transmitter power amplifier (PA) is programmable to seven power levels. Maximum output power is 10 dBm.

Figure 5. DPAC Technologies Airborne 802.11 module. It is fully Wi-Fi compatible. It is fast and very reliable over long distances especially at the lower 1, 2, and 5.5 Mbps data rates.

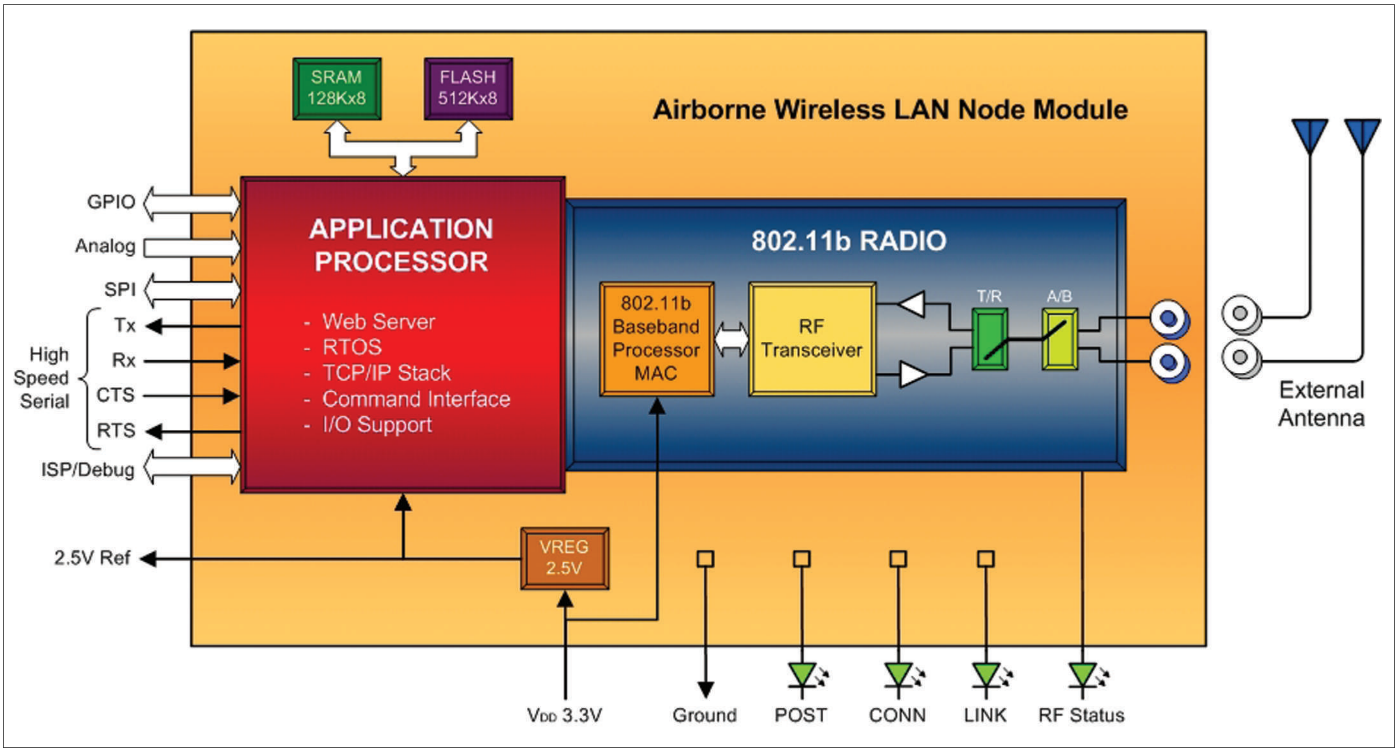


Figure 6. The Aerocomm ConnexLinks 900 MHz wireless modems for connectivity up to 20 miles with clear line of sight antennas.

The LNA input gives the receiver a -112 dBm sensitivity. This receiver is a direct conversion type, meaning the local oscillator PLL is set to the receive frequency producing a zero IF. To detect the FSK, dual mixers fed 90 degrees apart produce in-phase (I) and quadrature (Q) outputs that are filtered in active filters and a highly selective, switched capacitor, low pass filter. The cut-off frequency can be set to 100 kHz, 150 kHz, 230 kHz, or 340 kHz to match the chosen data rate. The filter output goes to the demodulator that recovers the serial binary signal.

Micrel also has a wide range of wireless chips that can be reviewed on their website at **www.micrel.com**

A really interesting new wireless product by Cypress Semiconductor is called WirelessUSB. It is a radio chip that incorporates a transceiver and is designed to work in the 2.4 GHz band, where one of the versions of ZigBee works. You could, in a way, call this chip ZigBee-lite, since it is not as fast and does not include all of the automatic networking capability, which is overkill for many applications.

There are two versions of the WirelessUSB — short range and long range. Both have a basic data rate of 62.5 Kbps. The short range device is good up to about 10 meters. It was optimized for things like wireless game controllers, computer mice, keyboards, and toys. The longer range device operates at up to a 50 meter range. It is ideal for many industrial applications, such as reading remote sensors.

Figure 4 shows a typical application, where multiple sensors send their data back to a central station for monitoring and control.

What makes this chip special is that it uses direct sequence spread spectrum (DSSS) with Gaussian FSK modulation. Spread spectrum is a far better communication method in noisy, industrial environments because it is more robust and tolerant of interference and multipath signal conditions, which are typical in such applications.

For more information, go to the website for the data sheets, app notes, etc. (**www.cypress.com**).

A really sophisticated wireless product is DPAC Technologies' Airborne radio modem. This is a complete IEEE 802.11b standard transceiver.

Also known as Wi-Fi, this is the technology used in laptops to link to the hot spots in airports, hotels, and other public places for Internet access. It is also widely used in home and office wireless LANs. This radio modem operates in the 2.4 GHz band with DSSS. The maximum data rate is 11 Mbps, but it can drop back to 5.5, 2, or 1 Mbps for longer range. Figure 5 shows the block diagram.

The applications processor supports lots of I/O types, including an eight channel analog multiplexer and A-to-D converter. The processor has a TCP/IP stack, making Internet access fast and easy. Also included is a web server on-chip that lets you monitor or control this device remotely via the Internet with any browser.

While this module is overkill for some industrial wireless applications, it is ideal for many uses, like

medical, automotive/trucking, and farming. Look at the website for greater detail — **www.dpactech.com**

Modules Make It Easier

Working with any of the wireless chips is a real challenge because of their incredibly small size. Yet, it can be done. You need some prototyping PC boards, like those made by Bellin Dynamic Systems. These are available from Jameco (**www.Jameco.com**).

Solder the chip to the board, then you can play with it like any other, larger IC. Breadboard sockets don't work too well with these chips because of the very high frequencies involved. You must keep all connecting leads very short and, if possible, use surface mount parts to ensure that.

Most of the chip vendors also have evaluation kits that include a prewired PC board with the wireless chip, antenna, and a microcontroller that can be programmed for any application.

These are inexpensive and really worth the money, given the time saved. Check out the manufacturers' websites for details.

If you just don't want to fool with the breadboarding part, you can go directly for the prewired module. Several companies that advertise in *Nuts & Volts* offer a wide range of low cost, prewired modules using these chips or their equivalents. Some of the companies offering a wide range of modules and related products include Abacom Technologies, Linx Technologies, Matco, MaxStream, Radiotronix, and Xemics.

For the ultimate in wireless links, you can go with a pair of transceivers, like Aerocomm's new ConnexLinks one watt 900 MHz transceiver (see Figure 6). With gain antennas high and a clear LOS orientation, the upper range is 20 miles. These units usually interface to a PC and use RS-232/RS-422/RS-485 serial data I/O with a rate up to 115.2 Kbps. A 2.4 GHz band unit is also available. See their website at **www.aerocomm.com** for more details.

There is one final thing about short range wireless products, like those covered here. The key to a successful application is the antennas. These can be a PC board pattern, a wire, or a commercial component. Even Yagis and verticals are used. Diversity antennas are really popular. The type and orientation of the antenna will determine if the radios talk to one other or not. I will address this critical antenna situation in a future column. **NV**

Advanced Techniques for Design Engineers

The Design Cycle

Serial Port Debugging

For some, debugging is the not-so-fun part of developing hardware and software applications. Whether you're designing complex subsystems for fighter jets or putting the finishing touches on that pet microcontroller project in your workshop, be certain that debugging will be part of your design cycle.

Debugging is an art and, in the true nature of art, the beauty of debugging lies in the eyes of the beholder. I have access to many of the latest and greatest high-dollar debugging tools. However, most of the time, I debug my codes and hardware using the cheapest tool in the lab — a standard RS-232 serial port. Take a look back at the June 2004 issue of *Nuts & Volts*. In that issue, I did a piece on digital filtering. Notice that the microcontroller-based digital filter hardware is supported by a serial port. Dig out your January 2004 issue of *Nuts & Volts*. There, you will find another article that I authored about PIC microcontrollers participating on an Ethernet LAN. Note that the Easy Ethernet Controller I wrote about in the January 2004 issue is also equipped with a serial port. In both the digital filter and Ethernet articles, the devices' serial ports worked double duty, providing both an application communications conduit and a firmware-controlled debugging interface.

I'm not the only programmer/designer hooked on the merits of serial port debugging. The first five chapters of Fred Eady's book, *Networking and Internetworking With Microcontrollers*, are dedicated to in-depth microcontroller serial port mechanics and they often appear in the firmware and hardware debug utilities that are included in the Ethernet code he details within the text of his book. Listing 1 is a code snippet from the Realtek RTL8019AS Ethernet driver described in Fred's book. The purpose of this code is to read all of the RTL8019AS's internal registers and display them using the Easy Ethernet device's serial port and a simple terminal emulator application, like HyperTerminal or Tera Term Pro.

The message here is clear. If your design permits, include a means to allow the host microcontroller simple RS-232 access to other devices and applications that can help

Listing 1. This code is called when it is desirable to know what is happening inside of the RTL8019AS register set. By employing the services of a simple serial connection, this code eliminated the need for additional LCDs, LEDs, and special debugging equipment. For those of you that want to see the rest of the driver code, you can download the complete Easy Ethernet driver from the *Nuts & Volts* website or the EDTP Electronics website.

```
//******************************************************************
//*   Read/Write for show_regs
//*   This routine reads a NIC register and dumps it out to the
//*   serial port as ASCII.
//******************************************************************
//
void readwrite()
{
     read_creg(i);
     bin2hex(byte_read);
     printf("\t%c%c",high_char,low_char);
}
//******************************************************************
//*   Displays Control Registers in Pages 1, 2 and 3
//*   This routine dumps all of the NIC internal registers
//*   to the serial port as ASCII characters.
//******************************************************************
void show_regs()
{
   write_creg(CR,0x21);
   cls();
   printf("\r\n");
   printf("    Realtek 8019AS Register Dump\n\n\r");
   printf("REG\tPage0\tPage1\tPage2\tPage3\n\r");

   for(i=0;i<16;++i)
   {
     bin2hex((int8) i);
     printf("%c%c",high_char,low_char);
     write_creg(CR,0x21);
     readwrite();
     write_creg(CR,0x61);
     readwrite();
     write_creg(CR,0xA1);
     readwrite();
     write_creg(CR,0xE1);
     readwrite();
     printf("\r\n");
   }
}
```

you in the debugging phase of your design cycle. With that thought in mind, let's look at how to go about RS-232 enabling your design.

Serially Enabling Your Hardware ...

Large memory and high pin count microcontrollers normally contain at least one USART (Universal Synchronous Asynchronous Receiver Transmitter). These days, it's becoming popular to include a pair of USARTs in the microcontroller's peripheral package. For instance, the PIC18F452 I used in the Digital Filter project and the PIC16F877 that drives the RTL8019AS-based Easy Ethernet Controller offer a single USART, while the high pin count PIC18F8621 I'm working with for a future *Nuts & Volts* offering is equipped with two EUSARTs (Enhanced Universal Synchronous Asynchronous Receiver Transmitter). The compiler guys are all over the multiple USART parts, as PicBasic Pro and Custom Computer Services PIC C offer dual-USART-ready BASIC and C compilers.

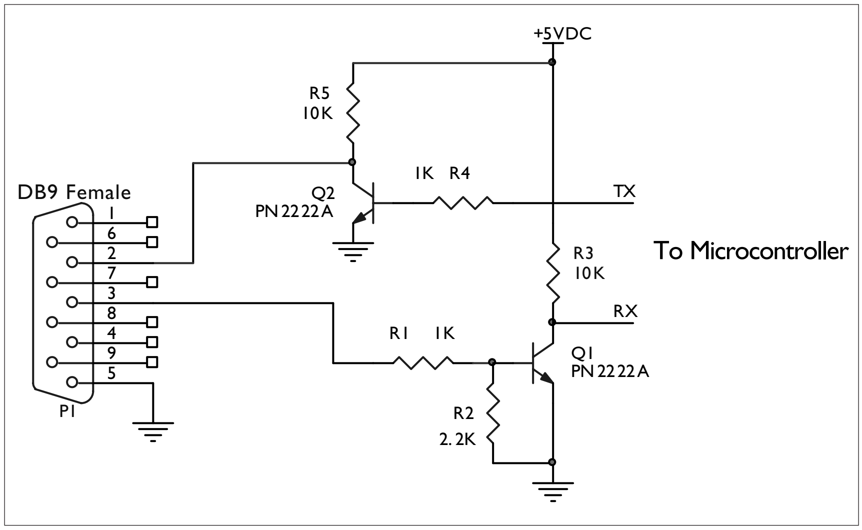


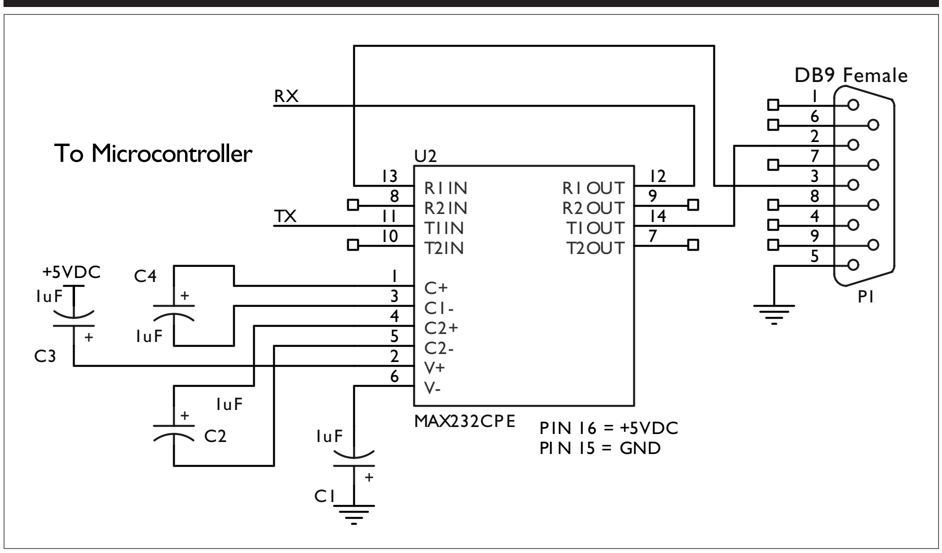
Figure 1. The only true RS-232 signal in this circuit is found at the base of transistor Q1. Q1 acts as a buffer/voltage translator converting the incoming RS-232 signal levels at its base to TTL levels at its collector. Q2 is a simple TTL-level switch. The transmit circuit works by swinging between near-ground and +5 VDC, which takes advantage of the minimum ±3 volt RS-232 voltage levels specified in the RS-232 standard. It is not necessary to drive the TX pin below ground as most true RS-232 interface ICs recognize the near-ground voltage as a mark.

Smaller microcontrollers — like the PIC10F and PIC12F series — don't house an internal on-chip hardware USART. On the smaller microcontrollers, there simply isn't enough silicon and I/O infrastructure to support a dedicated hardware USART. The good news is that a hardware USART is not a prerequisite for deploying serial port debugging. For those microcontrollers that don't contain an on-chip hardware USART, a software USART can be easily fabricated using microcontroller assembler coding techniques. Although coding a software USART in assembler is a viable alternative to a hardware USART, the advent of high quality, microcontroller-specific BASIC and C compilers eliminates the coding effort that would normally be required to build a suitable software USART from scratch using assembly language. Both of the PIC microcontroller compilers I'll talk about in this text contain easily deployed software USART functionality.

Before you write the first byte of serial port debugging code, you must make sure that the required microcontroller serial port hardware in your design is in place. Your serial port hardware can consist of a couple of transistors and resistors, as shown in Figure 1, or a specialized RS-232 IC, like the MAX232 circuit shown in Figure 2. The transistorized approach in Figure 1 is much less expensive to implement, but may be susceptible to line noise as true RS-232 levels are not used on the transmission cable. Because it's cheap, you'll find the dirty little transistor RS-232 circuit in quite a few commercially available electronic products.

Naturally, the application you choose for your microcontroller will dictate the hardware design and — if I was able to cover every possible hardware configuration

Figure 2. In the good old days, I used a pair of RS-232 interface parts that required positive and negative power supplies. Today, just add a few charge pump caps and a single positive supply voltage to get true RS-232 voltage levels from your microcontroller-supported serial port.

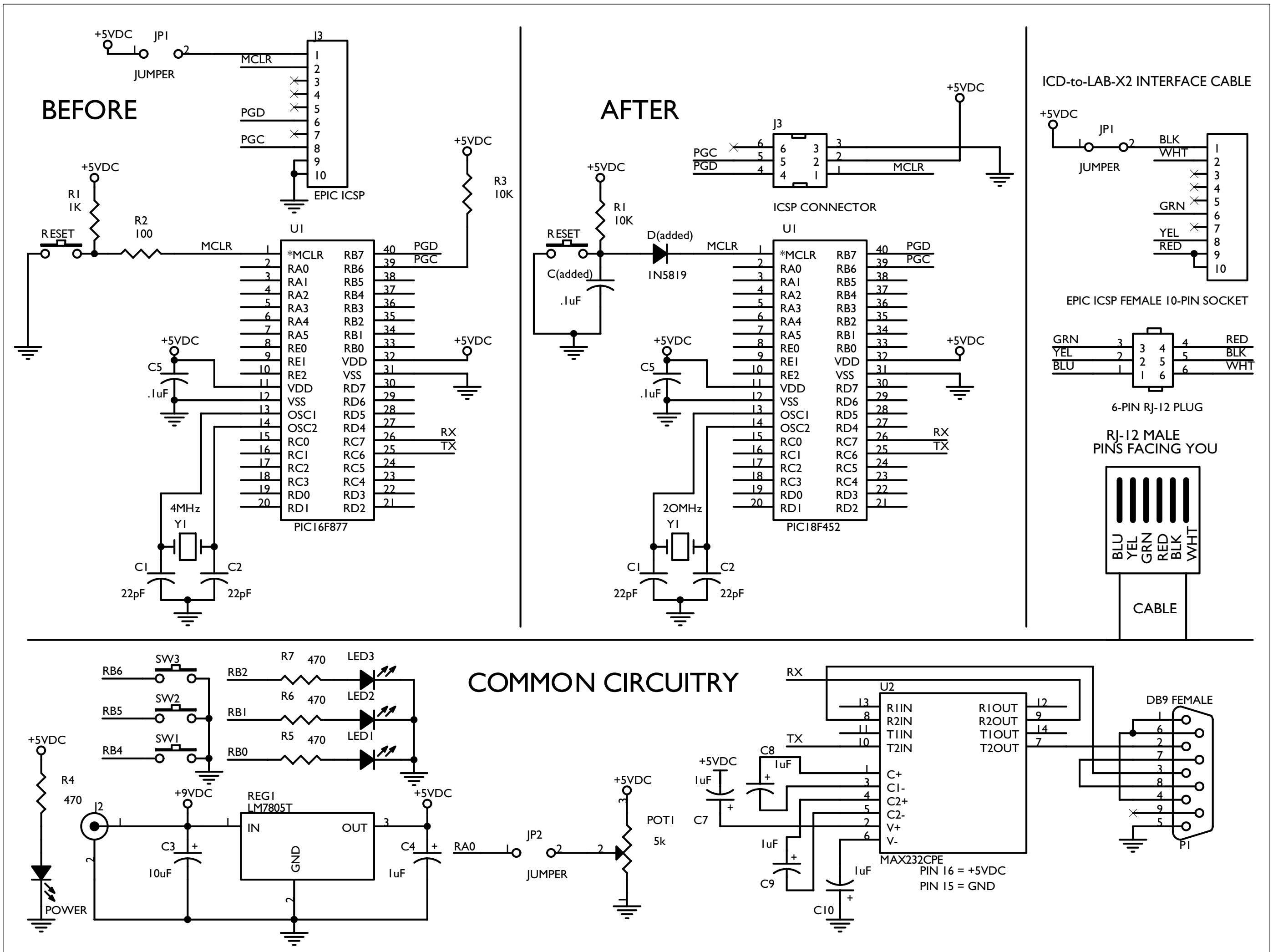


that you may have in mind — I would have most likely arrived on Earth in a spacecraft a very long time ago. Fortunately, I'm a true Earth boy and serial debugging techniques are not application dependent. That means that I don't need much hardware to show you some serial port debugging techniques. So, the hardware I've selected to use in this article to demonstrate the principles of serial port debugging is a very simple device that can be purchased from microEngineering Labs in the form of the LAB-X2 prototyping board. The LAB-X2 prototyping board is a basic 4 MHz implementation of the PIC16F876 and PIC16F877 microcontrollers. A standard RS-232 serial port — much like the one shown in Figure 2 — is included on the LAB-X2 board with power for the LAB-X2 components being provided by a wall wart and 7805 voltage regulator combination. Three LEDs, three pushbutton switches, and a potentiometer are attached to PORTB for your experimentation pleasure.

The LAB-X2's programming interface is designed around the EPIC programmer. The EPIC programmer is a fine PIC programming tool; however, I don't own one. So, for this discussion, I'll use Microchip's MPLAB ICD 2 to program the LAB-X2's PIC18F452. The MPLAB ICD 2 can be put into debugger mode to allow the target device on the LAB-X2 to be programmed very quickly. Using the MPLAB ICD 2 in debugger mode allows me to program, read, erase, start, stop, and reset the target PIC from the comfort of the MPLAB IDE without removing the PIC from the LAB-X2 socket.

When you get down to it, nothing is free. Payment for the convenience offered by the MPLAB ICD 2 and the MPLAB IDE must be made by the LAB-X2 hardware. To accommodate the MPLAB ICD 2, some minor hardware changes must be applied to the LAB-X2. I also wanted things to move about a bit faster than 4 MHz. So — while I was making mods to the LAB-X2's programming interface — I swapped in a 20 MHz crystal. The PIC18F452 and PIC16F877 are pin compatible. So, I dropped in a

Figure 3. There is absolutely nothing wrong with the LAB-X2 board out of the box. I'm just a frustrated hardware guy wearing coder clothing.



PIC18F452 instead of the PIC16F877 called out in the LAB-X2 schematic.

In addition to the mods on the LAB-X2 board itself, I had to fabricate an interface cable to allow the MPLAB ICD 2 to connect to the LAB-X2's 10-pin EPIC programming interface. A before and after view of the LAB-X2 is shown schematically in Figure 3, along with the pinout of the new LAB-X2/MPLAB ICD 2 programming cable. The bulk of the LAB-X2 mods were performed in the MCLR and PORTB program/clock pin areas. Basically, I eliminated the 10K pullup on the program clock pin (RB6) and the MCLR series resistor (R2). I replaced R2 with an SMT blocking diode, added a 0.1 µF manual RESET capacitor, and closed JP1.

The blocking diode insures isolation of the Vpp programming voltage when using the MPLAB ICD 2. At the same time, the voltage drop across the 1N5819 blocking diode is not great enough to prevent the MCLR pin from receiving the proper reset and run voltages. Shorting JP1 allows the MPLAB ICD 2 to receive power from the LAB-X2 power supply. The 0.1 µF RESET capacitor is optional and only comes into play when the RESET button is depressed. The MPLAB ICD 2 programming/debugging cable assembly is terminated on the MPLAB ICD 2 end with a standard male six-pin RJ-12 connector. The LAB-X2 end of the cable is fitted with a dual-row 10-pin female connector. The cable itself is a one-foot length of standard six-conductor flat cable that is also known as silver satin cable. A shot of my modified LAB-X2 and the MPLAB ICD 2 interface assembly can be seen in Photo 1.

After completing the LAB-X2 mods and rechecking my work, I applied power to the LAB-X2/MPLAB ICD 2 combination. Nothing appeared to be hot to the touch and the PIC18F452 and MPLAB ICD 2 weren't sending smoke signals, so I knew that the hardware was ready and now I could start writing the USART initialization code.

Enabling the USART ...

The very first coding task involves initializing the USART hardware in the LAB-X2's PIC18F452. Here's what the PicBasic Pro USART initialization code looks like:

```
define OSC 20   'define clock speed for the compiler
define HSER_RCSTA 90h          'binary 10010000
define HSER_TXSTA 24h          'binary 00100100
define HSER_BAUD 57600
```

PicBasic Pro assumes you are using a 4 MHz clock unless you specify otherwise. In the code snippet above, we have overridden the default clock rate and defined a 20 MHz microcontroller clock. The USART receiver module is activated by setting the SPEN (Serial Port Enable-bit7) and CREN (Continuous Receive

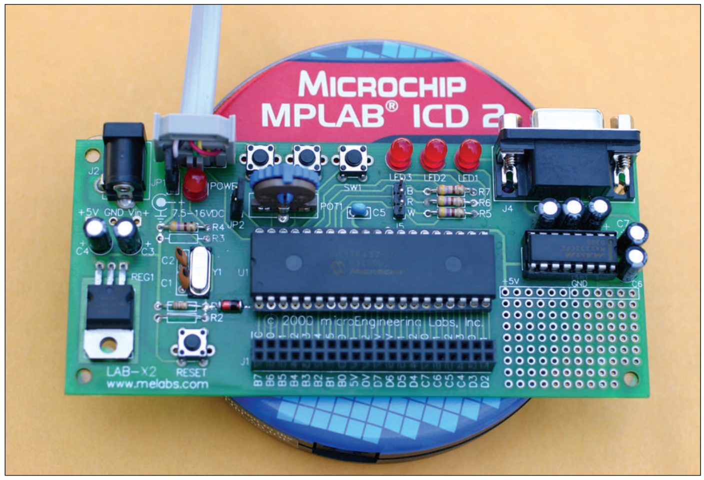


Photo 1. The LAB-X2 has just enough goodies to make it useful. The board works as designed right out of the box. I modified the board to allow the use of Microchip's MPLAB ICD 2.

Enable-bit4) bits in the RCSTA (Receive Status and Control) register. By not setting other bits in the RCSTA register, we set the asynchronous protocol to eight data bits, no parity.

The USART transmitter module is awakened by setting the TXEN (Transmit Enable-bit5) bit in the TXSTA

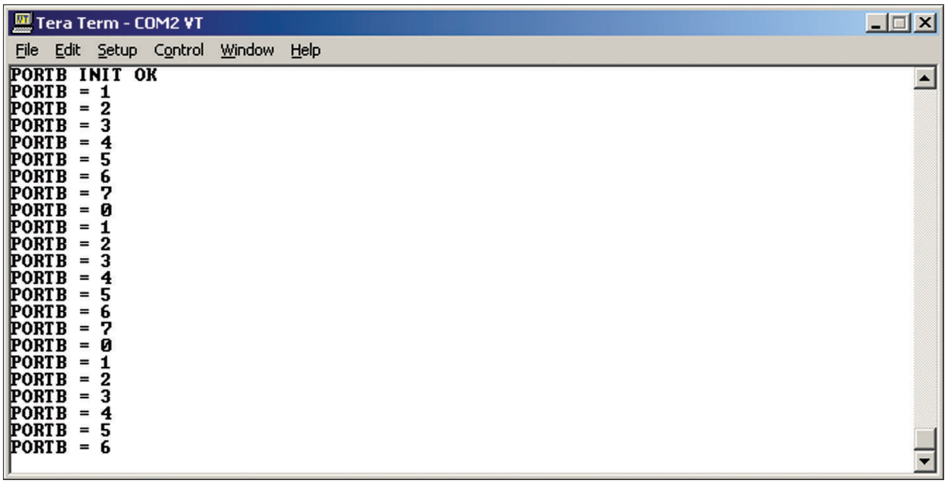


Figure 4. The count only cycles to 7, as the value of the Port B output latches that are not tristated by the TRIS function are being read. A maximum count of 255 is obtained when all of the Port B pins are designated as outputs and the MPLAB ICD 2 is disconnected from the LAB-X2.

(Transmit Status and Control) register. Since I have specified a baud rate of 57600 bps, the BRGH (High Baud Rate Select-bit2) bit must also be set to allow the USART baud rate generator to accommodate the high baud rate that I have selected. That's all it takes for PicBasic Pro to initialize the PIC18F452's hardware USART for 57600 bps operation.

Let's do the PIC18F452 USART initialization sequence using C and the Custom Computer Services C Compiler:

```
#use delay(clock=20000000)   //
define clock speed
                             // for the com-
piler
#use rs232(baud=57600,parity=N,xmit=
PIN_C6,rcv=PIN_C7,bits=8)
```

This C code does the same things to the same PIC18F452 USART registers that our PicBasic Pro BASIC USART initialization code did. The only difference in the two sets of code is that the C compiler does not assume that it is initializing a hardware USART, resulting in the requirement to explicitly call out the transmit and receive pins. The use of HSER definitions assumes that PicBasic Pro is working with a hardware USART. PicBasic Pro has special DEBUG and SEROUT commands that allow the programmer to send serial data from a preselected I/O pin without the need for the services of a hardware USART.

That was relatively painless. Let's put the PIC18F452's USART to work.

Checkpoints ...

Checkpoints are coded flags that can be inserted anywhere in your code. A checkpoint can be used to tell you many things about how and where your code is executing. For instance, a checkpoint can be used to verify that a routine actually executed. Another use for a checkpoint is to check for a loop condition in a module or routine. Let's write some PicBasic Pro code to initialize the LAB-X2 LED bank and throw in a couple of checkpoints:

```
chkpnt var bit
TRUE CON 1
FALSE CON 0
define OSC 20
define HSER_RCSTA 90h
define HSER_TXSTA 24h
define HSER_BAUD 57600

chkpnt = TRUE
TRISB = %11111000             ' Set PORTB.0-2 to outputs
PORTB = 0                     ' Turn off LEDs
if chkpnt = TRUE then
   Hserout ["PORTB INIT OK", 13, 10] ` checkpoint code
endif
loop:
    PORTB = PORTB + 1
```

```
if chkpnt = TRUE then
    Hserout ["PORTB = "]          ` checkpoint code
    Hserout [DEC PORTB, 13, 10] ` checkpoint code
    endif
    pause 1000
    goto loop
  End
```

In most cases, you won't want to run your checkpoint code all of the time. So, I added the variable chkpnt to act as a switch to either enable or disable the checkpoint code. Again, nothing is free and the payment for the chkpnt switch is additional code and additional testing time for the checkpoint IF...THEN statements. To save precious program area and to increase the speed of program execution, you'll definitely want to comment out all of your PicBasic Pro checkpoint code in your production firmware.

The checkpoint idea and final results are the same, but things are a bit different in C:

```
#use delay(clock=20000000)
#use
rs232(baud=57600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
#use fast_io(B)

#define chkpnt   // Comment line out for no checkpoint
code execution

void main(void)
{
        TRISB = 0b11111000;
        PORTB = 0;
        #ifdef chkpnt
            printf("PORTB INIT OK\r\n");
        #endif
        while(TRUE)
        {
            PORTB += 1;
            #ifdef chkpnt
                printf("PORTB = %u\r\n",PORTB);
            #endif
            delay_ms(1000);
        }
}
```

The Custom Computer Services checkpoint C code snippet produces the same results as our PicBasic Pro checkpoint code snippet (see Figure 4). The major difference in the BASIC and C code is that the C code snippet makes extensive use of C preprocessor directives and C include files. For example, the C definitions of TRUE and FALSE are predefined in the C include file 18F452.h.

Unlike my BASIC code, the #ifdef/#endif preprocessor statements don't compile into code and any code between the pair of preprocessor statements is also ignored by the C compiler if chkpnt is not defined. Don't sell PicBasic Pro short in this area as the #ifdef/#endif constructs are used extensively in the PicBasic Pro assembler library code.

For a programmer, nothing is free; the C code pays by not being as intuitive as the PicBasic Pro source code. For the benefit of those of you who don't do C, the #use fast_io(B) statement is a special Custom Computer Services C construct that simply turns off the automatic check of the specified port pins' TRIS status. Using the #use fast_io(B) statement allows me to control the port's TRIS status manually with the TRISB statement. Everything between the braces following the while(TRUE) statement executes in an endless loop, which is analogous to the loop:/goto loop construct used in the PicBasic Pro code. The PORTB+=1 statement increments PORTB by 1 and is equivalent to the PicBasic Pro statement PORTB = PORTB + 1. In fact, the += operator is a C shortcut and the increment PORTB statement can be written as PORTB = PORTB + 1 in both PicBasic Pro and C. Printf takes the place of Hserout with the /r and /n representing 13 (carriage return) and 10 (line feed) in the PicBasic Pro code. The %u inside the printf statement signifies that the value for PORTB (outside the quotation marks and behind the comma) should be displayed as an unsigned integer.

You can see from both the PicBasic Pro and Custom Computer Services C code examples that using checkpoints throughout your code can provide a quick visual indication of how and where your code is executing. Now that you're checked out on the hardware and understand the concept of checkpointing, let's look at a

Listing 2. This is all it takes to implement a firmware AND gate. A firmware NAND gate can be generated by simply changing the output_low statements to output_high statements and vice versa. I added comments behind the ending braces for those of you who aren't yet comfortable with C.

```
#define gate_output_pin  PIN_B2
void main()
{
        int8 logic_levels_in;
        set_tris_b(0b11111011);

        while(1){

        logic_levels_in = input_b();
        logic_levels_in &= 0x03;

        switch(logic_levels_in)
        {
                case 0x00:
                        output_low(gate_output_pin);
                        break;
                case 0x01:
                        output_low(gate_output_pin);
                        break;
                case 0x02:
                        output_low(gate_output_pin);
                        break;
                case 0x03:
                        output_high(gate_output_pin);
                        break;
          } //switch(logic_levels_in)
        }  //while(1)
}          //main
```

real world problem I encountered and solved using serial port debugging techniques.

Bait and Switch ...

Writing for *Nuts & Volts* is great in that I have the opportunity to touch, feel, and tell you about all of the latest and greatest microcontrollers and all of the stuff that supports them. However, there are some gotchas that come with the territory. Sometimes, things just don't work as designed.

For instance, Listing 2 is a simple firmware implementation of an AND gate. Port B pins 0 and 1 are the inputs to the AND gate with Port B pin 2 acting as the logical output of the AND gate. Port B is read continually (logic_levels_in = input_b();) and the lower two bits (the AND gate inputs) are isolated by logically ANDing the Port B input with a bit mask of 0x03 (logic_levels_in &= 0x03;). Physical AND gate logic works as follows:

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Listing 3. The As and the Zs were transmitted, but none of the putc statements inside the case constructs fired.

```
#define gate_output_pin  PIN_B2
void main()
{
        int8 logic_levels_in;
        set_tris_b(0b11111011);

        while(1){

        logic_levels_in = input_b();
        logic_levels_in &= 0x03;
        putc('A');       //added later
        switch(logic_levels_in)
        {
                case 0x00:
                        putc('0');
                        //output_low(gate_output_pin);
                        break;
                case 0x01:
                        putc('1');
                        //output_low(gate_output_pin);
                        break;
                case 0x02:
                        putc('2');
                        //output_low(gate_output_pin);
                        break;
                case 0x03:
                        putc('3');
                        //output_high(gate_output_pin);
                        break;
         } //switch(logic_levels_in)
        putc('Z');       //added later
        }  //while(1)
}          //main
```

The input logic levels represented by Input 1 and Input 2 are represented by the case statements in the switch(logic_levels_in) construct. For those of you who think in BASIC, the C switch statement works just like the Select Case statement in BASIC. The braces that surround the C switch construct are replaced by a Select Case/End Select statement pair in the BASIC implementation. In either language, the logic is identical. The output of our firmware AND gate only goes high when both of the inputs are logically high. This code is so simple that one would think that debugging would be totally unnecessary. Well, this seemingly simple piece of code did not run properly at all.

The physical circuit behind the code snippet in Listing 2 contains no visual aids, such as LEDs. So, instead of pulling out a scope or logic probe, I simply added putc (put character) statements inside of each of the case statements. The putc arguments added in Listing 3 match the case they are contained within.

For instance, if both inputs are logically low, the code in the `case 0x00:` construct would be executed, which would take the output of our firmware AND gate to a logical low state.

Notice that I've commented out each of the AND gate logical output states in the case constructs and replaced them with putc statements that tell me which case construct I have entered. I grounded both of the input pins and executed the program. Nothing was sent to the terminal emulator. Okay ... I then tied both of the inputs high. Still, nothing showed up on the terminal emulator screen. I decided to write a small serial routine to make sure that data could indeed be sent by the PIC. The code was very simple:

```
while(1)          //loop forever
{
     putc('A');          //send A from the
serial port
     delay_ms(500);      //wait for half a
second
}
```

A string of As spurted across the terminal emulator screen when I ran the little piece of test code. That told me the hardware was working as designed. I then put a putc('A') before the switch code and a putc('Z') following the switch code. The A and the Z both printed in sequence, telling me that the code was executing from top to bottom. However, I still did not see any of the case numbers being transmitted. I finally came to the conclusion that the switch construct was not running and turned to the assembler listing that the C compiler generated for some clues.

Listing 4 is a part of the assembler code

generated by the C compiler. The switch logic begins at program memory address 0037. Code at program memory addresses 0037 and 0038 test the incoming logic level data from Port B, looking for a 0. If the incoming data is determined to be 0 (both firmware AND gate inputs at logic low levels), code at program memory address 0039 is executed and a jump to the case 0x00: construct code is executed. Code beginning at program memory address 003A tests for a 1, code beginning at address 003E tests for a 2, and so forth.

If a match is made, code execution jumps to the appropriate case construct and executes the instructions inside the selected case code construct. After code in each of the case constructs is executed, a "break" out of the switch construct is performed with a GOTO 057 instruction. This insures that only one of the case constructs will execute for each pass through the switch code structure. Code at program memory address 0057 sends the PIC's program counter back to the beginning of the never-ending loop, which starts at program memory address 0033.

All of the switch code operates using data memory address 0x08. Note that data at data memory address 0x08 must be 0, 1, 2, or 3 if any of the case construct code is to be jumped to and executed. Therefore, data residing at data memory location 0x08 must be greater than 3 for the switch construct code execution to be bypassed. The number entering the switch code at program memory address 0037 cannot possibly be greater than 3, as our logic level input is trimmed to two bits by ANDing the logic level input data from Port B with 0x03 (binary 00000011). The logical binary numerical possibilities for the two least significant bits are 00, 01, 10, and 11. So, what gives here?

Sources

Custom Computer Services
Custom Computer Services C Compiler for the PIC Microcontroller Family
www.ccsinfo.com

microEngineering Labs
LAB-X2
PicBasic Pro
www.melabs.com

EDTP Electronics, Inc.
Easy Ethernet Devices
RTL8019AS Driver Code
Networking and Internetworking with Microcontrollers
www.edtp.com

Listing 4. Normally, I don't go to this level of debugging, but, when simple stuff doesn't work, all of the clues reside here. If you wonder about how tight the code your compiler generates is, this is the place to go.

```
....................
....................           while(1){
....................
....................           logic_levels_in = input_b();
0033:  MOVF    06,W
0034:  MOVWF   0B
....................           logic_levels_in &= 0x03;
0035:  MOVLW   03
0036:  ANDWF   0B,F
....................
....................           switch(logic_levels_in)
....................           {
0037:  MOVF    08,F
0038:  BTFSC   03.2
0039:  GOTO    047
003A:  MOVLW   01
003B:  SUBWF   08,W
003C:  BTFSC   03.2
003D:  GOTO    04B
003E:  MOVLW   02
003F:  SUBWF   08,W
0040:  BTFSC   03.2
0041:  GOTO    04F
0042:  MOVLW   03
0043:  SUBWF   08,W
0044:  BTFSC   03.2
0045:  GOTO    053
0046:  GOTO    057
....................                   case 0x00:
....................                          putc('0');
0047:  MOVLW   30
0048:  MOVWF   0E
0049:  CALL    003
....................                          //output_low(GP2);
....................                          break;
004A:  GOTO    057
....................                   case 0x01:
....................                          putc('1');
004B:  MOVLW   31
004C:  MOVWF   0E
004D:  CALL    003
....................                          //output_low(GP2);
....................                          break;
004E:  GOTO    057
....................                   case 0x02:
....................                          putc('2');
004F:  MOVLW   32
0050:  MOVWF   0E
0051:  CALL    003
....................                          //output_low(GP2);
....................                          break;
0052:  GOTO    057
....................                   case 0x03:
....................                          putc('3');
0053:  MOVLW   33
0054:  MOVWF   0E
0055:  CALL    003
....................                          //output_high(GP2);
....................                   break;
0056:  GOTO    057
....................            }
....................           }
0057:  GOTO    033
.................... }
....................
```

Can you spot the problem? Things go wrong beginning at program memory address 0033, which reads the logic levels of the Port B pins and stores the byte in the W register. Program memory address 0034 moves the incoming logic level byte to data memory location 0x0B. Code at program memory addresses 0035 and 0036 perform the AND mask function that clips off six of the most significant bits that were read from the pins of Port B. Notice that the incoming data that was ANDed is stored back into data memory location 0x0B by code at program memory address 0036 (ANDWF 0B,F).

Our switch logic works with data in data memory location 0x08. The data needed by the switch construct is actually stored in data memory location 0x0B. That means that the data at data memory location 0x08 is indeterminate and can range randomly between 0x00 to 0xFF. Since data memory location 0x08 is untouched by our ANDing process and our switch code is not running, the value held in data memory location 0x08 must be greater than 3. Any value greater than 3 will fall through all of the tests performed by the switch construct's code.

Thus, the bug is in the assembler code generated by the C compiler. We smashed this bug without having to employ test equipment or building up temporary debugging hardware.

Changing the 0x0B to 0x08 at program memory locations 0034 and 0036 allowed the switch construct code to act on the true incoming logic level data from Port B, which is now stored where the switch code expects to find it: in data memory location 0x08. Once I patched the C compiler's output assembler file, the corresponding putc statements allowed me to easily debug the code by jumpering the firmware AND gate inputs and testing all four of the binary AND gate input combinations. No scopes, no logic probes, no in-circuit emulators, and no hassle — that's the beauty of employing serial debugging in your design cycle.

Applying the Technique ...

I think you've got the idea. Adding a minimal amount of RS-232 hardware to your design enables serial debugging checkpoints in interrupt routines, subroutines, macros, and just about any other piece of code you want to keep tabs on. Serial debugging is what you make it. It can be as complicated as Fred's RTL8019AS register code or as simple as a putc statement in my firmware AND gate code. **NV**

AMATEUR RADIO & TV

BATTERIES/CHARGERS

BUYING ELECTRONIC SURPLUS

CCD CAMERAS/VIDEO

CIRCUIT BOARDS

COMPONENTS

COMPUTER

Hardware

Microcontrollers / I/O Boards

Software

DESIGN/ENGINEERING/ REPAIR SERVICES

EDUCATION

ENCLOSURES

EVENTS

KITS

LASERS

MISC./SURPLUS

PROGRAMMERS

PUBLICATIONS

RF TRANSMITTERS/ RECEIVERS

ROBOTICS

SATELLITE

SECURITY

STEPPER MOTORS

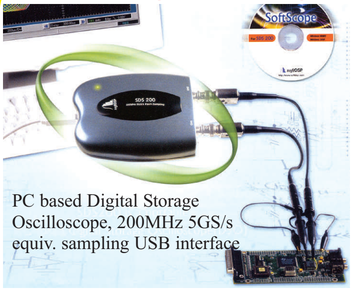
TELEPHONE/CELLULAR

TEST EQUIPMENT

TOOLS

WIRE/CABLE & CONNECTORS

Digital Storage Oscilloscope Module



PC based Digital Storage Oscilloscope, 200MHz 5GS/s equiv. sampling USB interface

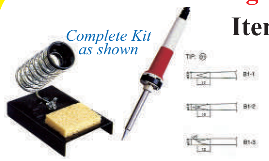
Convert any PC with USB interface to a high performance Digital Storage Oscilloscope. This is a sophisticated PC basedscope adaptor providing performance compatible to mid/high level stand alone products costing much more! Comes with two probes.

Details & Software Download at Web Site

> Test Equipment
> Oscilloscopes/Outstanding Prices

Item# 200DSO ..Only$859.00

40 Watt Soldering Kit



Item# ZD200B

Only $9.95 !

A 40 watt soldering iron with a 3-wire power cord, fast heating ceramic element and a grounded tip to protect static sensitive devices. *Comes complete with a soldering stand and three replacement soldering tips*, which include a conical tip, chisel tip and a 45° chisel tip.

Details at Web Site

> Soldering Equipment & Supplies

13.8V, 6A DC Regulated Power Supply

new!

Item# CSI1862

Ideal for mobile tranceivers, high amp stepper motors and CNC machines.

Details at Web Site

$24.95

> Test Equipment
> Power Supplies

Innovative 5 in 1 DMM

Integrated Sound/ Light/Humidity Sensors

Item# CSI8209

*Alt-Mag sweep
*1mV/Div vertical sensitivity
*Alternate trigger
*X5 sweep magnification

Details at Web Site

$49.00

> Test Equipment
> Digital Multimeters/World Beater Prices

Protek 60MHz Dual Trace Scope

Super Blowout Price!

*60MHz Bandwidth
*Dual Channel
*Alternate Trigger
*Autofocus
*Large 6" CRT
*Alt-Mag sweep for simultaneous display of normal & X10 trace
*Sweep speeds to 10nS/Div.
*10kV acceleration voltage
*Internal sync separator circuit for stable triggering of video signals

Item# 6506

$349.00 !

While Supplies Last!

Details at Web Site

Sold by others for Over $700!

> Test Equipment > Oscilloscopes/Outstanding Prices

Circuit Specialists 20MHz Dual Trace Scope

Item# CSI6502

*20Mhz Bandwidth
*Alt-Mag sweep
*1mV/Div vertical sensitivity
*Alternate trigger
*X5 sweep magnification
*Large 6"CRT/autofocus
*Comes w/2 (x1 & x10) probes)

$269.00 !

Details at Web Site

> Test Equipment > Oscilloscopes/Outstanding Prices

3M™ DataCom Cable Tester

UNBEATABLE PRICE

Limited Time Offer

Item# DT-2000

Only $49.00

This unit allows for mapping, testing and troubleshooting of various lines, including installed data communications, phone wiring and coaxial cable runs.Performs multiple test on the following cable types, up to 1000 feet in length: Unshielded telephone cables with RJ-11 and RJ-45 connectors; Ethernet 10 (100) Base-T; Token Ring; EIA/TIA-568 A/B; AT&T 258a; USOC; 50 or 75 ohm Coax with F or BNC connectors.

Includes: Holster, Case, 7 Remotes & Telecom Alligator Clips

Details at Web Site > Test Equipment > Specialty Test Equipment

RF Field Strength Analyzer

New Fantastic Low Price: $1299!

Compare at Over $2000 !

The **3201** is a high quality hand-held RF Field Strength Analyzer with wide band reception ranging from 100kHz to 2060MHz.The 3201 is a compact & lightweight portable analyzer & is a must for RF Technicians. Ideal for testing, installing & maintenance of Mobile Telephone Comm systems, Cellular Phones,Cordless phones, paging systems, cable &Satellite TV as well as antenna installations.May also be used to locate hidden cameras using RF transmissions

Item# 3201

*WFM/NFM/AM/SSB modulated signals may be measured.
*Signal Levels up to 160Channels can be displayed simulaneously on the LCD
*PLL tuning system for precise frequency measurement and tuning
*Built-in Frequency Counter
*LED Backlight LCD (192x192 dots)
*All fuctions are menu selected.
*RS232C with software for PC & printer interface
*Built-in speaker

(Includes Antenna)

(Limited Offer)

Details at Web Site > Test Equipment > RF Test Equipment

Programmable DC Electronic Load

Item# CSI3710A

Only $349.00!

A programmable electronic DC load capable of supporting up to 150W of power. Can be used with supplies up to 150VDC and 30A. It features a rotary selection switch and a numeric keypad used to input the maximum voltage, current and power settings. Perfect for use in laboratory environments and schools, or for testing DC power supplies or high-capacity batteries. It also features memory and can be connected to a PC, to implement remote control and supervision.

KEY FEATURES:
*Maximum admissible power: 150W
*Maximum current rating: 30A
*Maximum voltage input: 150V
*Maximum voltage and current settings can be adjusted in 10mV/10mA increments
*Storage for 10 different voltage/current settings
*Monitored by PC software
*Can be used in a parallel connection

Details at Web Site

> Test Equipment > Power Supplies

FC5001 2 Way FM Radio Tester/ FC6002 Radio Frequency Tracer

The **FC5001** 2-way FM radio tester has the ability to lock automatically and almost instantly on to any FM signal within its frequency range. The **FC6002** radio frequency tracer is useful in locating stuck transmitters or bugging devices in a room or automobile. It excels at silent detecting RF signals for RF security and counter-surveillance applications.

FC5001: $99.00 < RF Security > FC6002: $149.00

Details at Web Site > Test Equipment > RF Test Equipment

PROGRAMMABLE DC POWER SUPPLY

Only $199.00 !

Item# CSI3645A

*Stores up to 10 settings for fast & accurate recall
*Backlit LCD display
*High Resolution (1mV)
*PC compatible (with optional RS-232 adaptor module)
*Easy programming w numeric keypad or fast rotary code switch
*Power shut down memory function

PDF Manual available at CircuitSpecialists.com

Details at Web Site > Test Equipment > Power Supplies

BAG of LEDs DEAL

Normal brightness LEDs now available in RED, GREEN or YELLOW in 3mm or 5mm sizes.Your choice. *Each bag* contains 100 of the *same* LEDs.

BAG-RED3MM..............$1.50 **BAG-RED5MM.............$1.50**
BAG-GREEN3MM........$1.50 **BAG-GREEN5MM.......$1.50**
BAG-YELLOW3MM....$2.00 **BAG-YELLOW5MM...$2.00**

Super Bright LEDs Deal

53B3SCS08...5mm Blue SB LED(1500max MCD)1+ $0.70 /10+ $0.65 /100+ $0.60
5G3UTB-2... 5mm Green SB LED(1100max MCD)1+ $0.45 /10+ $0.35 /100+ $0.30
5R3UT-2/R...5mm Red SB LED(3500max MCD) 1+ $0.25 /10+ $0.20 /100+ $0.15
53BW3SCC08..5mm White SB LED(3500max MCD)1+ $1.69 /10+ $1.49 /100+ $1.18
5Y3STC-2....5mm Yellow SB LED(3500max MCD) 1+ $0.25 /10+ $0.20 /100+ $0.15

Details at Web Site > Semiconductor Devices > LEDs, Displays & Lamps

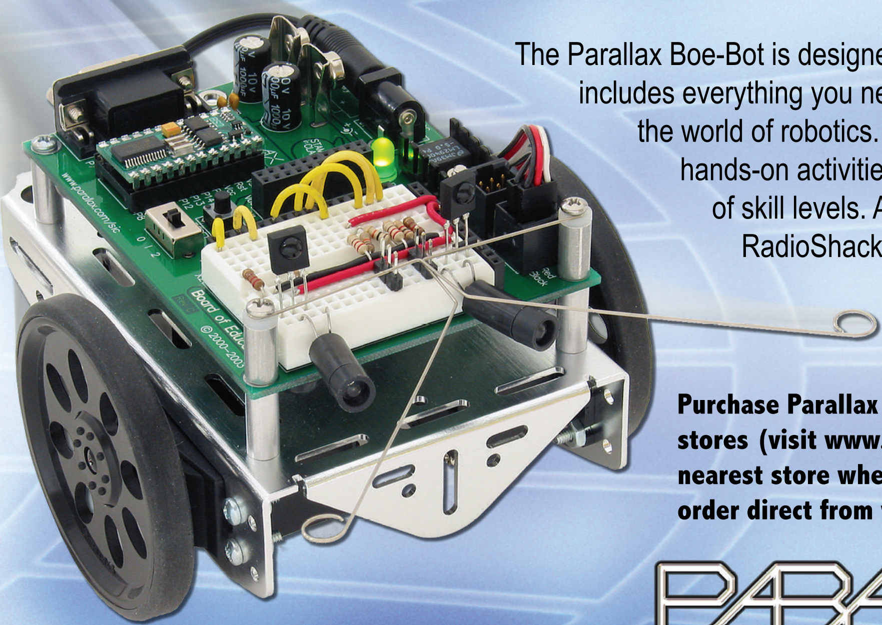
Visit our website for a complete listing of our offers. We have over 8,000 electronic items on line @ **www.CircuitSpecialists.com**. PC based data acquisition, industrial computers, loads of test equipment, optics, I.C's, transistors, diodes, resistors, potentiometers, motion control products, capacitors,miniature observation cameras, panel meters, chemicals for electronics, do it yourself printed circuit supplies for PCb fabrication, educational D.I.Y.kits, cooling fans, heat shrink, cable ties & other wire handleing items, hand tools for electronics, breadboards, trainers, programmers & much much more! *Some Deals you won't believe!*

Circuit Specialists, Inc. 220 S. Country Club Dr., Mesa, AZ 85210
800-528-1417 / 480-464-2485 / FAX: 480-464-5824

Circle #106 on the Reader Service Card.

Boe-Bot™ robots now at RadioShack®

RadioShack and Parallax have teamed up. Demonstrating their commitment to the hobbyist, experimenter and student, RadioShack now offers Parallax in select retail stores and online at www.RadioShack.com. You can now buy our BASIC Stamp What's a Microcontroller? Kit and the Boe-Bot Robot Kit through RadioShack.



The Parallax Boe-Bot is designed to be user-friendly. It includes everything you need to launch yourself into the world of robotics. The Boe-Bot offers 40+ hands-on activities suitable for a wide range of skill levels. And it is expandable with RadioShack component parts.

Purchase Parallax products at select RadioShack stores (visit www.RadioShack.com to find the nearest store where this product is offered) or order direct from www.parallax.com.

PARALLAX

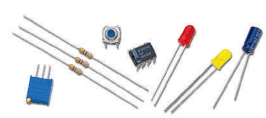
www.parallax.com

Boe-Bot and the Parallax logo are trademarks of Parallax, Inc.

Limited-Time Offer!

RadioShack.

50% OFF Most Parts and Components*



Offer Valid 9/30/04 – 10/28/04

RadioShack would like to reward the hobbyist, experimenter and student with an offer of 50% off parts and components to fulfill your project needs.

* Vaild on items in our Parts and Components drawers. Quantities limited to stock in store only. Products may not be available in all stores. No rain checks. This offer may not be combined with any other offers or discounts (no employee discounts). Offer valid at all participating RadioShack stores. See stores for more details.

Circle #154 on the Reader Service Card.